University of Alberta

Image Compression and Quality Assessment: An Investigation Exploring Wavelet Packets and Human Visual System Characteristics

by

Nicholas Andrew Roeder ©

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Master of Science**.

Department of Computing Science

Edmonton, Alberta
Fall 1997

0-612-22665-4

University of Alberta

Library Release Form

Name of Author: Nicholas Andrew Roeder

Title of Thesis: Image Compression and Quality Assessment: An Investigation Exploring Wavelet Packets and Human Visual System Characteristics

Degree: Master of Science

Year this Degree Granted: 1997

Nicholas A. Roeder

Nicholas Andrew Roeder
205 Hillcrest Place
Edmonton, Alberta
Canada T5R 5X6

Date: June 27, 1997.

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **Image Compression and Quality Assessment: An Investigation Exploring Wavelet Packets and Human Visual System Characteristics** submitted by Nicholas Andrew Roeder in partial fulfillment of the requirements for the degree of **Master of Science**.

Xiaobo Li

Dr Xiaobo Li

Hong Z

Dr Hong Zhang

Dr Stuart Jackson

Date: June 25, 1997.

Abstract

This thesis investigates the applicability to image compression of a flexible wavelet decomposition in conjunction with a simple human visual system (HVS) model. Several strategies for a computationally-light, top-down, wavelet packet decomposition are developed and compared. Techniques are examined for modeling three primary HVS sensitivities. Gamma correction is investigated to model luminance sensitivity. An anisotropic model of the the contrast sensitivity function is used to design a preprocessing filter. Uniform and non-uniform quantizers are designed based on HVS frequency sensitivity experiments and the HVS masking effect, respectively. Quality assessment of the reconstructed image is examined, emphasizing the importance of applying quantitative *and* qualitative assessment mechanisms. Compression results are reported on a large variety of images. The performance of this scheme is found to be competitive or yield better visual results than other wavelet-based techniques on many images. With certain image classes, this flexibile scheme can yield significant improvements in quality.

Acknowledgements

This project could not have been completed without the assistance and encouragement of many people. I would like to give my heartfelt thanks to the following people:

- To Dr Xiaobo Li, my thesis supervisor, for your long-time support through this project, three summer projects, and three courses. Thanks for seeing potential in me when I had only doubts.
- To Dr Andrew Bradley, a mentor to me in this project, for your creativity, good humor, understanding, generosity, encouragement, patience, flexibility, and muffin recipes.
- To my friends and colleagues Jason Knipe, Pawel Jachowicz, and Wladek Olesinski for your conversation, ideas, commiseration, and frequent squash breaks.
- To the Voûte Vereniging for your generous scholarship support.
- To Dr Joseph Gee, Randy Hirsch, and Johane Sinclaire for doing what you do so well.
- To my family, my parents Antoinette and Michael and my sisters Sasha and Rita, for your infinite support, strength, and patience.
- And lastly and most importantly to my partner and soul-mate, Peggy, for your support, understanding, patience, conversation, zaniness, and love.

Thank you all!

Contents

List of Tables

List of Figures

List of Acronyms and Abbreviations

ADF	acceptable distortion factor
bpp	bits per pixel
CCIR	International Radio Consultative Committee
CD-ROM	compact disc read only memory
codec	compressor-decompressor
compandor	compressor-expandor
CSF	contrast sensitivity function
DCT	discrete cosine transform
EZW	embedded zero trees of wavelet coefficients
FFT	fast Fourier transform
GIF	graphics interchange format
HVS	human visual system
JPEG	Joint Photographic Experts Group
LBG	Linde, Buzo, and Gray
LVQ	lattice vector quantization
MSE	mean-square error

PGM	portable graymap
PQS	picture quality scale
PSNR	peak signal-to-noise ratio
QT	quadtree
R-D	rate-distortion
SPIHT	set partitioning in hierarchical trees
TSVQ	tree-structured vector quantization
VDP	visual difference predictor
VQ	vector quantization
WMSE	weighted mean-square error
WP/HVS	wavelet packets and human vision system characteristics
WPSNR	weighted peak signal-to-noise ratio

Chapter 1

Introduction

1.1 The Increasing Applicability of Digital Images

The use of digital images within computer software is becoming increasingly prevalent. Modern multimedia programs are being developed which contain hundreds or perhaps thousands of images. sometimes requiring more memory space than that available on a single CD-ROM. Many medical (e.g.. CAT scans) and industrial applications require the analysis of digital images. The explosive growth of the World Wide Web has also propagated the use of digital images to a much wider audience.

Since digital images are composed of hundreds of thousands of pixels, each pixel needing a certain amount of memory space, data for the entire image can grow to significant amounts (i.e.. upwards of 1 megabyte). As the number of images also grows, more disk space is required for their storage, and the time to transmit these images across a network also increases. The ability to *compress* the images to take up significantly less space and transmission time would be valuable.

Anyone who has browsed the World Wide Web with a standard personal computer modem connection can relate to long download times associated with large images. However, with respect to storage capacity, it appears that hard disk space is rapidly plummeting in cost. Is it really valuable to have the capability to compress images in order to conserve storage space?

Some applications involve the acquisition of large quantities of digitized images that have been previously stored as photographs or on paper. The Federal Bureau of Investigation (FBI) in the United States, for example, has over 200 million fingerprint records stored in the form of paper cards. Additionally, this database is growing by approximately 30 thousand new fingerprints every day [5]. Having the fingerprints in electronic database form could dramatically increase the efficiency of file searching.

Once the fingerprint cards have been digitized, the database of uncompressed images would require on the order of *thousands of terabytes* of storage space. Disk space of that magnitude is still fairly costly.

1.2 Redundancies in Images

In order to significantly reduce the size of an image file, a compression scheme in which some information is irreversibly lost is needed. Without losing information. existing compression techniques can achieve approximately 2:1 to 5:1 compression ratios. Methods which approximate the image by losing some data in the process can achieve from 10:1 to 30:1 compression while maintaining decent image quality. There are certainly some cases, perhaps involving legal issues associated with a medical application, in which losing information would not be acceptable, even if the compressed image "looks the same" as the original. However, the majority of applications involving digital images can accept images which are not visibly different from the original uncompressed image. The main problems in designing an image compression scheme are determining how best to *approximate* image data which is less visually significant. and determining *which* data *is* less significant.

Fortunately, most natural images have been found to contain a great deal of redundant information. Traditionally, the redundancies have been classified into three types [28]:

1. **Interpixel Redundancy.** Significant regions of an image may contain pixels of *similar* color. For example, a landscape scene may have a large expanse of clear blue sky. It should not be necessary to record each individual pixel of such a region. Even without large expanses of unchanging color, most images have significant correlation between pixels. Digitizing an image by sampling it at the minimal Nyquist rate produces small interpixel changes on average, leading to high correlation.

2. **Coding Redundancy.** In uncompressed images, each pixel is typically represented by a fixed-length code. The inherent redundancy in this representation can be removed by using shorter codes for more likely pixel values, or colors. This strategy is not unlike the well-known Morse Code, in which frequently used letters (e.g., "e") are represented with much shorter codes than those infrequently used (e.g., "z").

3. **Psychovisual Redundancy.** The human eye is not capable of discerning all possible colors which can be produced by modern computer hardware. The amount of available colors or gray levels can be reduced without significantly impacting the visual quality of an image.

Items 1 and 2 above have been well studied. Many researchers agree that aspects of the human visual system (HVS) need to be more thoroughly examined and incorporated into image compression schemes. This examination may lead to the discovery of more psychovisual redundancies (item 3 above) in images than with traditional techniques. Experiments have found that the HVS has variable sensitivity to luminance ("brightness") level, the frequency of an image signal, and to the actual content of the image.

1.3 Image Quality

Since image compression usually involves the loss of some information, each compression technique should be judged on the quality of the image it produces, as compared to the original image. The final user of a compressed image will be the best judge of its quality, but this person is not always available for comment when developing new image compression methods. Even if a user were available, each would have her own subjective opinion of image quality. Developing experiments involving many human observers is time consuming and expensive. Traditionally, quantitative analysis has been applied to examine the amount of errors or distortion in the reconstructed image, but one number representing image quality does not indicate where the distortion occurs or the degree of perceptible distortion.

Again, aspects of the HVS can be incorporated into judging image quality. Quantitative measures can weight errors in different parts of the image based on their content to calculate a more representative number. Additionally, distortion "maps" can be created which show exactly *where* the errors are in the image. If these maps are developed involving HVS characteristics, they can even indicate whether or not the distortion is visible to an "average" user.

1.4 Project Scope and Motivation

The work presented in this thesis is part of a larger project involving several researchers at the University of Alberta, Canada, working with a grant from Motorola Canada.

The purpose of the project is to develop efficient and secure image compression technologies which can be used to transmit images over wireless networks to and from portable computers or hand-held video equipment.

Any new technique developed during this project should be elegant and *computationally simple*. Desktop computers are constantly becoming more powerful, but hand-held units do not offer the same computational speed, nor the same memory capacities.

It is possible that certain users would require the ability to transmit images securely, preventing any person but the indicated recipient from viewing the image. Since wireless networks are publicly accessible, *encryption* technology will have to be used, so that any potential "intruder" would have to attempt to unscramble the image information. One of the main thrusts of the Motorola project is investigating the combination of image compression and encryption. However, we will not directly deal with encryption herein, since other members of our research team have already examined this issue [11].

Another goal of the project is to design compression schemes which are capable of supporting *progressive transmission*. That is, the viewer should be able to see an approximation of the final image, which constantly becomes more precise as more compressed data is downloaded. Some compression schemes require the user to download the image file in its entirety before viewing is possible.

We will be attempting to design a new image compression system which is computationally light, incorporates a model of the human visual system, and is amenable to both progressive transmission and encryption.

1.5 The Approach

Recently, so-called *wavelet* mathematical techniques have been demonstrated by many researchers to achieve high performance in image compression, by transforming the image pixel information into coefficients of compact basis functions. Few researchers have attempted to take advantage of the power of wavelets coupled with certain aspects of what is known about the workings of the human visual system. We propose that more research and experimentation is required into this coupling and the interactions between the two methods.

In this thesis, we investigate the applicability to image compression of a flexible wavelet decomposition (i.e., wavelet packets) in conjunction with a simple human visual system model. We also examine image quality issues using both quantitative and qualitative analysis. Before building and testing a final image compression system, we examine many different parameters and configurations of such a system in an attempt to find the "optimal" system. This analysis is thorough, in that a relatively large number (54) of images from a variety of applications is used.

The final compression scheme developed within is potentially computationally light, flexible and adaptable to image signal content, user-friendly, and more suitable to HVS sensitivities than many standard techniques. The performance of our scheme is found to be competitive or yield better visual results than other wavelet-based techniques on many images. With images of certain types, the flexibility of our scheme can yield significant improvements in quality, assessed both quantitatively and qualitatively.

1.6 Outline

What follows is an organizational outline of the remainder of the thesis.

Chapter 2. Here we present a general overview of image compression and quality assessment techniques, with emphasis on wavelet methods and human visual system characteristics.

Chapter 3. A newly developed image compression framework is presented, involving wavelet packet decomposition and human visual system modeling. We call this system a *WP/HVS framework* due to the fact that several processes are optional or have more than one implemented approach.[1] This framework is designed to facilitate independent testing of and experimentation with the compression components.

Chapter 4. The components and parameters of the WP/HVS framework are analyzed in detail in order to design the "best" possible image compression system. Within this chapter, we introduce and apply a relatively large, categorized, experimental image library.

Chapter 5. This chapter takes the newly developed WP/HVS image compression system and compares its performance on images of many types with other systems available today. The results are analyzed both quantitatively and qualitatively.

[1] WP/HVS is an abbreviation for wavelet packets / human visual system characteristics.

Chapter 6. The thesis is concluded with a comprehensive summary, a list of specific contributions to the image compression body of knowledge, and several proposals for further research.

Chapter 2

An Overview of Image Compression

2.1 Introduction

Image compression is an extremely active area of signal processing research. This chapter presents a general overview of image compression systems and techniques, including:

- a model image compression system;
- specific methods relating to each component of the model with emphasis on wavelet compression techniques;
- characteristics of the human visual system and its relation to image quality assessment and new directions in compression research;
- and a brief mention of several image compression industry standards.

2.2 Codec Model

In order to maximally reduce the redundancies found in images, most compression schemes involve the combination of several processes. The system which is formed by this combination is often called a *compressor-decompressor* or *codec*. Figure 2.1 depicts the simple codec model which will be described here.

Original Image. The image to be compressed, denoted $f(x,y)$, is input into the system. Normally this is described as a sequence of bytes, which, in the case of standard grayscale images, yields a possibility of $2^8 = 256$ intensities. Color images require extra bytes to represent chrominance information. Most compression techniques (including those in this thesis) are developed using grayscale

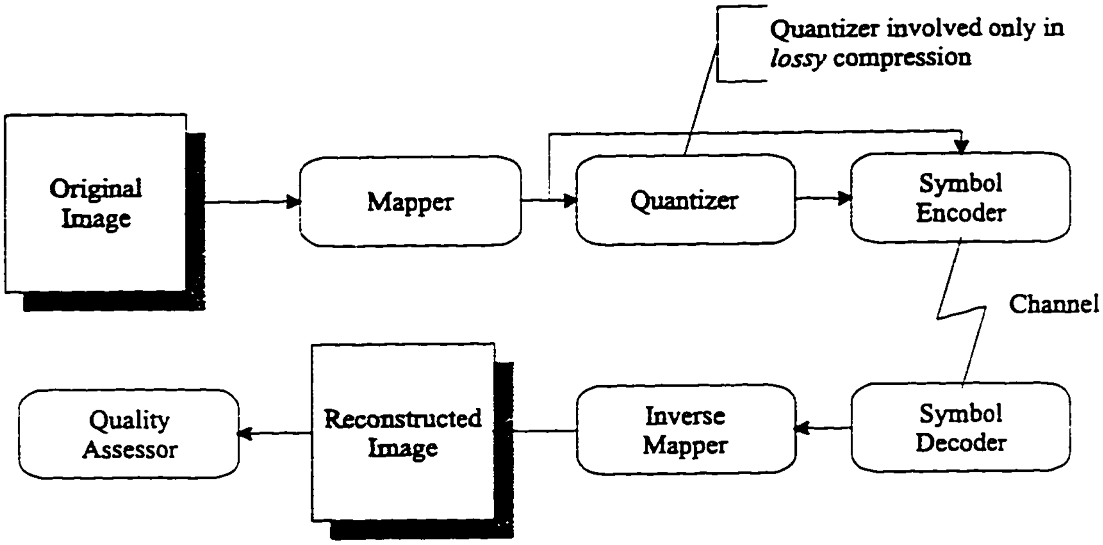


Figure 2.1: Block diagram of the image codec model.

images. It is easy to extend grayscale codecs to compress color images by treating the chrominance information separately.

Mapper. The intensity correlation among neighboring pixels in an image can be significant, depending on the geometric structure of the objects within the image. The correlation is termed *interpixel redundancy* since the value of a single pixel can be reasonably predicted from its neighbors. The mapper is used to perform a reversible transform on the pixels to a non-visual format, decorrelating the pixels as much as possible. This transform may be performed in the *spatial domain*, directly calculated from the image pixels, or in the *transform domain*, mapping the pixels to transform coefficients of basis functions.

The amount of information in an image signal can be measured by the *entropy*. The theoretical minimum codeword length can be estimated by a first order approximation of the entropy in bits per pixel (bpp),

$$H = -\sum_i p(i) \log_2 p(i) \text{ bpp}, \tag{2.1}$$

where i is the pixel intensity and $p(i)$ is the probability of intensity i. Larger entropies are associated with higher information content and, therefore, lower redundancy. If it were equally probable to find each intensity in a grayscale image (i.e., the intensities had a uniform probability distribution), the entropy would be 8 bpp. The main purpose of the mapper is to modify the intensity distribution to be more peaked, decreasing the entropy of the signal.

Quantizer. The human visual system (HVS) perceives certain visual stimuli as less important than others. To achieve significant image compression, the *psycho-visually redundant* information can be removed by quantization (i.e., representation with less precision). Quantization should be accomplished with minimal loss in perceived quality, since it is not reversible. Codecs incorporating quantization are termed *lossy* since some information from the original information is permanently lost.

Note that *lossless* codecs (i.e., without quantization) will not be dealt with here. The majority of compression research is done in the area of lossy codecs, since lossless schemes can only achieve around 2:1 compression [13]. Lossy codecs can often achieve 20:1 or greater compression ratios.

Symbol Encoder. In most images, the occurrence probabilities of different intensities vary, especially after passing through the mapper. This trend can be used advantageously by applying a variable length code, assigning longer codes to low probability symbols, and shorter codes to higher probability symbols. This reversible process reduces the *coding redundancy* and is often termed *entropy coding* since the goal is to have an average code length near the entropy.

The two most popular coding techniques are Huffman coding and arithmetic coding (see [13], Appendix 1). Huffman designed a code in 1952 which assigns a variable length code to each symbol based directly on that symbol's probability. Arithmetic coding represents a string of symbols by a single codeword indicating an interval on the real number line. Both coding methods can approach the entropy of a source, but arithmetic coding often has superior performance and can be adaptive with varying source statistics.

Entropy coding schemes will not be examined in more detail, due to the proven success of the standard techniques.

Channel. Images are usually compressed in order to take up less storage space, to reduce required transmission bandwidth, or both. The channel in the model codec represents offline storage or a network. In the specific scope of the Motorola research project, the channel is a publicly accessible wireless network. It may be desirable in such a situation to *encrypt* the compressed image to facilitate privacy of the image transmission. The portable computers deployed in conjunction with the wireless network will have relatively small storage capacities, making it especially important to have high compression rates.

Symbol Decoder and Inverse Mapper. Once the compressed image needs to be viewed again, either by retrieving it from storage or from across the network, the symbol decoder and inverse mapper convert the codes back to visual information. The variable length entropy code is translated to the quantized symbols by the symbol decoder. These quantized values are then transformed back into pixel information by the inverse mapper.

Reconstructed Image. The reconstructed image is reproduced from the now decompressed data. This image is usually denoted $\hat{f}(x, y)$ to show that it may not be identical to the original (i.e., lossy compression was used).

Quality Assessor. The quality of the reconstructed image is often assessed by the user in order to determine its acceptability in a particular application. Quality assessment is an important component of image compression research, because some measure is needed to determine if a new method yields any improvement in performance. The assessor may range from qualitative judgments of humans to a quantitative mathematical calculation leading to a quality number scale.

A quantitative measure is preferable, since qualitative human trials can be costly and time consuming. Experiments involving humans may also be unintentionally affected by environmental factors like ambient light, quality of the display device, vision accuracy of the human subject, or even the emotional state of the subject. Unfortunately, it is very difficult to develop a quantitative measure which truly reflects image quality as agreed upon by a substantial proportion of the population.

The mapper, quantizer, and quality assessor are the main focus of this thesis. The remainder of this chapter will examine these components in more detail.

2.3 Spatial Domain Mapping Techniques

There are many established techniques for mapping the original pixels into some form more amenable to compression. As previously stated, some methods are applied in the spatial domain, in which the original pixels can be directly manipulated.

2.3.1 Predictive Coding

Perhaps the history of image compression began in 1950, with the patent of predictive coding [13]. The idea was born from the observation that many images contain sequences of similar intensities, i.e., interpixel redundancy.

Historically, images have been scanned, transmitted, and displayed in a linear left to right, top to bottom fashion. The coder can therefore predict (or estimate) the value of the current pixel from those previously seen in a neighborhood to the left and above. In general, the predicted value $\hat{S}(i)$ of the current pixel $S(i)$ is a weighted linear combination of past pixels,

$$\hat{S}(i) = a_1S(i-1) + a_2S(i-2) + \ldots + a_nS(i-n), \tag{2.2}$$

where n is the *order* of the predictor or the count of past pixels. The prediction error

$$e(i) = S(i) - \hat{S}(i) \tag{2.3}$$

is then coded. The probability of small errors is usually high, which leads to a very "peaked" distribution of symbols to encode. This is the precise goal of the mapper as described in Section 2.2. The errors can then be quantized, yielding lossy compression.

In an image, there is no true "causal" relationship of previously seen pixels to the current pixel. Some researchers have looked into non-causal predictive coding, in which both the "past" and "future" neighboring pixels are used to predict the current pixel [67]. Using future pixels adds theoretical and implementation complexity, since future pixels are not available when decoding. An iterative decoding procedure can be used to increase the precision of the values at each iteration (i.e., the future pixels can be initially assumed to be the mean of the input image).

Basic predictive coding by itself cannot compete with the performance of more modern lossy methods, compressing down to about 2 bpp [13], but it is still the standard technique for lossless coding. The Graphics Interchange Format (GIF) is a lossless image compression standard which originated with CompuServe and is based on predictive coding [45]. GIF is often used to encode small icons and images on the World Wide Web.

2.3.2 Segmentation-Based Compression

Given a certain minimum quality constraint, images with low detail content can be compressed more than those containing large amounts of intensity changes (i.e., edges). This simple fact has lead researches to what is dubbed "second generation" or image content-based compression [34]. Th:s approach is based on the premise that the future of image compression lies in taking advantage of the content and structure of the image.

From one point of view, the image can be segmented into regions according to the amount of detail content. This step could be considered as part of the mapper in the codec model of Figure 2.1 since its goal is to reduce the interpixel redundancies. The segmentation can be extracted and recorded separately. Aside from this structure, the remaining information describes the fine details of each region, which can be compressed using traditional ("first generation") compression methods, such as predictive or transform coding. Figure 2.2 depicts the second generation image compression process.

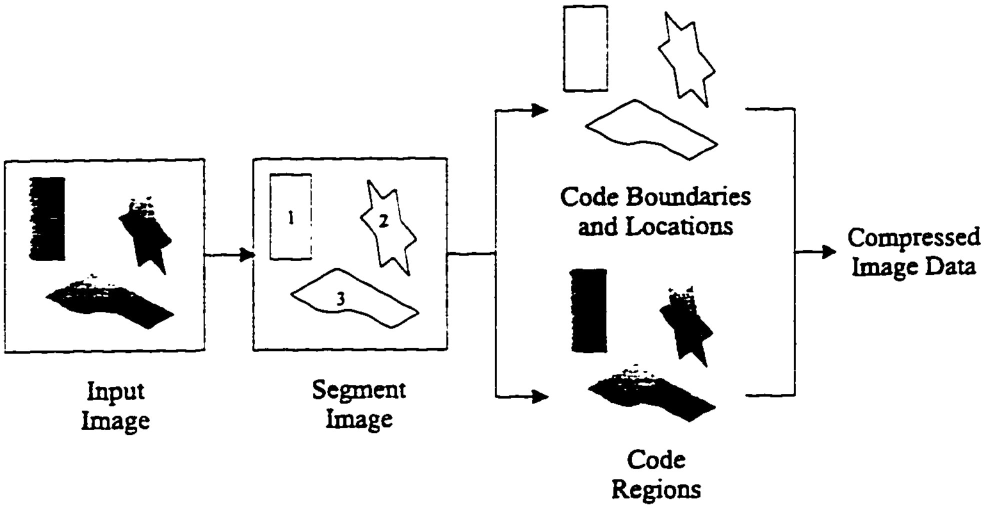


Figure 2.2: Segmentation-based image compression.

In order to efficiently store the image segmentation, a data structure must be designed that can minimize the space required while maximizing the amount of stored image content information. There are many different techniques for decomposing or segmenting an image into non-overlapping spatial regions. Each method has associated with it a certain amount of "overhead" information which describes the structure of the segmented image. Several of these techniques will now be described.

Quadtree Segmentation

The amount of extra information required is, in general, proportional to the level of segmentation and the amount of detail in the individual image to be compressed. The quadtree (QT) method takes into account the image content, albeit in a small amount, by recursively decomposing an image into four sub-images if a homogeneity measure is not met [61, 16].

El-Sakka and Kamel looked into developing a better splitting criterion for QT, using the mean-square deviation from a neighborhood mean [21]. This facilitated segmentation into smooth, textured, and edge blocks in order to allow the allocation of more bits to perceptually important areas. Vaisey and Gersho used a QT splitting technique and coded smaller, more perceptually important blocks with a classified vector quantizer (see Section 2.5.3), and larger "random texture" blocks with a transform coder [56]. This computationally intensive algorithm produced high quality results in the 0.35–0.7 bpp range. In general, the highly regular structure of QT is extremely easy to store in a very small space, but the decision to split each sub-block is inflexible. This does not necessarily treat the image content in a realistic way.

Some modifications to QT have been explored to provide a more flexible segmentation approach in which siblings in the tree may be merged if they meet certain criteria [24]. This technique effectively increases the descriptive power of the QT by allowing non-square (i.e., rectangular) segments. However, it was found that on real images this extra descriptive power had little overall effect on compression efficiency because of the increased overhead required for storage of the modified quadtree.

Polygonal Segmentation

Another approach that examined alternatives to QT found that the power of using a more adaptive segmentation scheme, i.e., using rectangles or polygons, also did not offset the additional cost required for the new data structure [14].

Wu and Fang developed a method which recursively partitions an image into polygons whose edges can be oriented horizontally, vertically, at 45^0, or 135^0 [62, 63]. This information is stored in a binary tree. The authors proposed that the method is particularly suitable to progressive transmission, since the polygons can store more structure information than squares. According to the authors, "Compression ratios from 100:1 to 20:1 were obtained for fair to excellent image reproductions." The authors decline to quote any quantitative measure, perhaps due to the addition of Gaussian white noise to the polygons which is added to increase the realism.

Irregular Segmentation

More extreme segmentation-based methods involve detecting highly irregular regions in images. Cortez *et al.* examined some techniques for using morphological operations in split-and-merge segmentation [17]. The boundaries of these regions are often coded using efficient contour coding [41]. Once the boundaries are represented, the region interiors can be coded using predictive coding, a transform coder, vector quantization [55] (see Section 2.5.3), or texture modeling and polynomial expansions [35].

Challenges

Second generation compression schemes require that an image be intelligently segmented into its component parts, and that these segments be modeled with extremely high accuracy and reliability. With the current "state of the art" in image-understanding technology, Ryan *et al.* proposed that second generation methods would not be feasible in the foreseeable future [50]. Second generation methods involve an intrinsically high computational overhead, especially if the segmentation and modeling is to be both reliable and accurate.

To obtain good compression rates, image content-based methods require prior knowledge of the image content. This knowledge facilitates minimization of the model information to be transmitted to the receiver, since it has prior indication of what kind of model to use, e.g., a face model. The knowledge-based approaches have been predictably termed "third generation" image compression. In general, when the image scene is unknown, or when the image data is a poor fit to the model, these methods yield poor performance. In other words, these techniques are unable to effectively represent general, unmodeled objects. In addition, the boundary coding techniques can yield a fairly artificial-looking image, since the boundaries and regions are compressed in such different ways, leading to objectionable contours and borders.

The strong performance of wavelet compression, a transform domain mapping technique, makes it difficult for second generation methods to be competitive [13]. This fact has inspired some researchers to incorporate wavelet decomposition into second generation coding techniques [50]. However, Ryan *et al.* found that texture modeling in the wavelet domain on polygonal segmentations only worked effectively on highly textured images. Stand-alone wavelet techniques are also computationally light and require no image understanding or segmentation.

In conclusion, as more complex segmentation schemes are used, the improvement in compression performance usually does not compensate for the increased overhead of boundary encoding. For this reason, our research will concentrate on wavelet compression, which will be discussed in Section 2.4.2.

2.4 Transform Domain Mapping Techniques

Thus far we have only discussed mappers which deal directly with the image pixels. Another mapper architecture involves a linear reversible transform, mapping the pixels into transform coefficients in order to alter the distribution of symbols. For most natural images, significant portions of higher spatial frequencies (see Section 2.4.2) will have very small coefficients, and can be coarsely quantized or discarded without noticeable image degradation.

A transform will represent the signal as a combination of basis functions. The Fourier transform, for example, uses sine and cosine basis functions. If the input signal is a single frequency, the output will be a single coefficient of the basis function corresponding to that frequency. Multiple input frequencies produce multiple coefficients. In a 1D case, the transform coefficients $\mathbf{C}$ can be found by

$$\mathbf{C} = \mathbf{T}_1\mathbf{X}, \tag{2.4}$$

where $\mathbf{T}_1$ is the basis matrix and $\mathbf{X}$ is the original signal data vector. In order to retrieve the reconstructed image, the inverse transform,

$$\mathbf{X} = \mathbf{T}_2\mathbf{C}, \tag{2.5}$$

is used, where $\mathbf{T}_2 = \mathbf{T}_1^{-1}$. In order to extend the transform to the 2D case of images, the rows can be transformed first, followed by the columns.

Two important aspects of transforms should be noted here: *orthogonality* and *orthonormality* [13]. Orthogonality refers to the fact that each coefficient can be found independently of all others. The transforms should be orthonormal, that is energy conserving, so that the energy in the original data may be rearranged but will equal the total energy of the transform coefficients. The rearrangement is important, however, since it modifies the symbol distribution, allowing more efficient quantization and coding.

The two most popular transforms used in image compression are the discrete cosine transform and the wavelet transform.

2.4.1 Discrete Cosine Transform

The discrete cosine transform (DCT) was developed in the early 1970s [13]. It has been found to be far more efficient in representing real images than the discrete Fourier transform, and has the added benefit of not using complex numbers. DCT uses only cosine basis functions and has been used extensively in image compression.

The Joint Photographic Experts Group (JPEG) developed a standardized specification for an image compression system based on the DCT, quantization, and a variable length code [57]. Small 8×8 sub-blocks are first transformed using the DCT. Larger sub-blocks can decrease the distortion as measured by the mean-square error (MSE, see Section 2.6), but also increase computation demand exponentially. The JPEG standard uses a custom 8×8 quantization table which is designed to take into account which coefficients are perceptually more important than others. JPEG is widely used in lossy compression applications, including the World Wide Web.

2.4.2 Wavelet Transform

Like the Fourier transform, the wavelet transform represents input signals in terms of basis functions, however these functions are more complex than sines and cosines [20]. Since sine and cosine functions are infinite in support, the Fourier transform, in theory, requires the integration of a signal over infinite time and frequency. As a result, an isolated impulse signal results in an infinitely wide energy distribution in the Fourier domain.

It has been found that wavelet coefficients can more efficiently code image signals than Fourier coefficients, particularly high-contrast areas like edges, due to the compact-support wavelet basis functions (i.e., "short filters").[1] Another benefit is that the blocking artifacts associated with DCT methods like JPEG are not evident using wavelet transforms, since there is no need to split the image into smaller sub-blocks before transforming. The wavelet decomposition has been shown to be similar in some respects to the functionality of the human visual system [36] and is very well-suited to progressive transmission, as will be shown.

Compression schemes based on the wavelet transform appear to be moving into direct competition with industry standards. The FBI standard for compressing fingerprint images uses the wavelet transform [6]. Analog Devices (a company that produces digital signal processors) is reportedly producing a wavelet codec chip for

[1] See [47], pg. 604 for a direct comparison of the representation power of wavelet versus Fourier coefficients.

only $50 [12].

We will now present some background theory on wavelets, report on several research developments in wavelet compression, and describe a modification called wavelet packets.

Sub-band Coding and Wavelet Theory

This section presents a brief summary of wavelet theory and is not intended to be complete. For more detailed overviews see [13, 47].

In general, most images have areas of quickly varying signals and areas of slowly varying signals. The rate of signal variation can be described in terms of *spatial frequency*. Images often have very strong signals in their low-pass spectra which contain very little redundancy, but the high-pass signals can be highly repetitive. These facts have inspired a technique called *sub-band* coding, in which the image signal is filtered into several different frequency bands to facilitate efficient coding of areas of varying spatial frequency, and to concentrate compression effort in high-pass bands. Using a mapper of this structure will result in several different, peaked distributions of coefficients.[2]

If the image were to be filtered into a set of separate sub-bands, the number of data elements would multiply by the band count. It turns out, however, that the individual sub-bands may be sub-sampled in proportion to the reduction of detail from the filtering operation. Signal processing theory states that any signal can be reconstructed accurately as long as it is sampled at greater than the Nyquist rate [25]. The Nyquist rate is twice the highest possible spatial frequency. Sub-sampling allows the total number of elements in all sub-bands to remain the same as the original image. The reconstruction operation at the decoder uses interpolation to reverse the effect of sub-sampling. Transform coding techniques like Fourier and DCT are really specific cases of sub-band coding where one coefficient is used to represent an entire sub-band (i.e., they do not incorporate spatial localization).

The wavelet transform is a type of sub-band coding system which uses basis functions with highly compact support, which are also scalable and translatable, facilitating multiresolution decomposition.[3] The wavelet transformation matrix $\mathbf{T_1}$ (recall equation (2.4)) holds the coefficients of the filters L and H. L is the low-pass filter which outputs a "smoother" version of the image and H is the high-pass filter which

[2] Significantly, the inspiration for sub-band coding appears similar to that of second generation coding (see Section 2.3.2).

[3] Note that given the fixed input image resolution, only *lower* resolutions can be generated.

generates a detail signal.[4] The detail signal represents the differences between the smooth images at adjacent levels of resolution.

The matrix can be constructed so that low-pass filtering, high-pass filtering, and sub-sampling are done in one matrix multiplication. As a simple example of a wavelet basis function we consider one developed by Daubechies [47]. This wavelet consists of only four coefficients,

$$c_0 = (1+\sqrt{3})/4\sqrt{2}, \tag{2.6}$$

$$c_1 = (3+\sqrt{3})/4\sqrt{2}, \tag{2.7}$$

$$c_2 = (3-\sqrt{3})/4\sqrt{2}, \tag{2.8}$$

$$c_3 = (1-\sqrt{3})/4\sqrt{2}. \tag{2.9}$$

Applying a transformation matrix of the form (where blanks represent zeros)

$$\begin{bmatrix}
c_0 & c_1 & c_2 & c_3 & & & & & & & \\
c_3 & -c_2 & c_1 & -c_0 & & & & & & & \\
 & & c_0 & c_1 & c_2 & c_3 & & & & & \\
 & & c_3 & -c_2 & c_1 & -c_0 & & & & & \\
\vdots & \vdots & & & & & \ddots & & & & \\
 & & & & & & & c_0 & c_1 & c_2 & c_3 \\
 & & & & & & & c_3 & -c_2 & c_1 & -c_0 \\
c_2 & c_3 & & & & & & & & c_0 & c_1 \\
c_1 & -c_0 & & & & & & & & c_3 & -c_2
\end{bmatrix} \tag{2.10}$$

to an input data vector performs a convolution and sub-sampling operation. The results from the filters are sub-sampled 2:1 (the Nyquist rate) which yields the wavelet coefficients—the final output of the mapper. The coefficients can then be quantized, symbol encoded, and sent to the channel.

Reconstruction of the image is accomplished by interpolating the sub-sampled coefficients and then using the inverse of $\mathbf{T}_1$, normally being equal to $\mathbf{T}^t$ since the matrix is designed to be orthogonal.

The entire wavelet compression process is represented in Figure 2.3. The wavelet transform has been described as a 1D operation, but like other transforms, 2D images can be transformed by rows first and then columns, resulting in low-pass and high-pass sub-bands in both directions at each level.

The decomposition algorithm can be recursively repeated on the low-pass result, giving a multi-resolution, pyramidal decomposition. Figure 2.4 represents the result

[4] Since the filtering involves a convolution operation, the sides of the image can cause problems due to a lack of data. Symmetric extension or "padding" of the image can eliminate this problem.

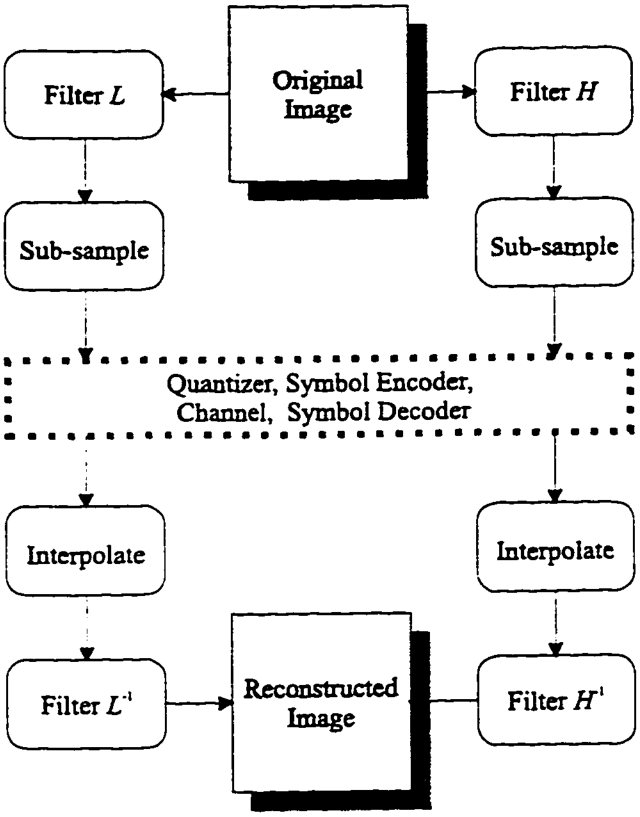


Figure 2.3: Block diagram of a wavelet transform.

of a 2-level wavelet decomposition on an image. In the figure, each sub-band is labeled with a letter pair representing the horizontal (H) and vertical (V) band-pass result. Additionally, the symbol $S(\lambda, \theta)$ indicates the level λ and orientation θ of each sub-band. The result is one low-pass representation of the image with several high-pass frequency sub-bands containing peaked distributions of wavelet coefficients. This image representation contains the same number of elements as the original image.

The decomposed image is particularly well-suited to progressive transmission due to its multi-resolution nature. The sub-bands may be transmitted from lowest- to highest-resolution sub-band, while the decoder reconstructs each sub-band to form an image which becomes progressively more precise.

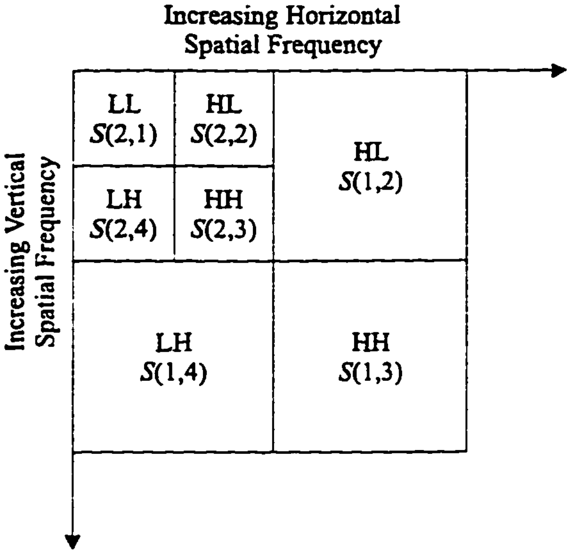

Increasing Horizontal Spatial Frequency
Increasing Vertical Spatial Frequency
LL S(2,1)
HL S(2,2)
LH S(2,4)
HH S(2,3)
HL S(1,2)
LH S(1,4)
HH S(1,3)

Figure 2.4: Representation of a 2-level wavelet decomposition of an image.

Wavelet Compression Techniques

Wavelet compression is a very active area of image compression research and many techniques have been developed based on the standard wavelet transform. Due to the multi-resolution nature of a wavelet decomposition, the sub-bands at adjacent levels in the pyramid have a parent-child relationship, with each coefficient in one resolution having four child coefficients in the next higher resolution, and so on. Lewis and Knowles used this relationship to build a tree structure of coefficients [36]. It was found that insignificant coefficients at a low resolution had a high correlation with insignificant coefficients in higher resolutions. Recognizing this relationship, an entire branch in the tree of coefficients can be sent by one zero element to indicate that no coefficients in that branch need to be coded due to their insignificance.

The tree-structured idea was furthered by Shapiro to yield an embedded zerotree wavelet algorithm (EZW) [53]. This technique requires the encoder and decoder to agree on a specific transmission order of the coefficients from more significant to less significant. The embedded nature of the coded image allows the decoder to stop retrieving coefficients at any point to yield an image of a certain detail, which facilitates flexible progressive transmission. Said and Pearlman enhanced the EZW algorithm to use set partitioning in hierarchical trees (SPIHT) [51]. In addition, the general idea of EZW has been placed in a rate-distortion optimizing framework (which, of course, adds computational complexity) in order to produce the minimum quantitatively-measured error for a given bit-rate [64].

Other compression schemes using the wavelet transform involve vector quantization of the coefficients [3, 33] or texture modeling [50]. Some researches have also examined the use of human visual system characteristics (HVS) in conjunction with wavelet compression, which will be discussed in detail below.

All of the above algorithms apply the wavelet pyramid decomposition scheme, which has been shown to work well on most natural images. But what about images which exhibit an unusual spatial-frequency distribution, in which more information may be contained in particular sub-bands? A more flexible and adaptive scheme might *dynamically* choose which sub-bands should be decomposed at run-time, rather than always decomposing the LL sub-band. Such a technique exists and is called *wavelet packets* [15, 48, 65].

Wavelet Packets

The pyramid-structured wavelet transform is a specific case of the wavelet packet transform in which only the LL sub-band is decomposed at each level. The packet transform allows the decomposition of any sub-band and, therefore, also requires that the decomposition structure be recorded. Perhaps the most sensible way to represent the decomposition is with a quadtree. An example decomposition and its corresponding quadtree are shown in Figure 2.5.

It has been expressed that variations on sub-band splitting techniques have an insignificant impact on image compression according to bit-rate and quantitative error measures [13]. Nevertheless, some people have examined wavelet packet decomposition with the hypothesis that some types of images might benefit from its use [15, 48, 65].

Xiong *et al.* have extended their work in wavelet compression [64] to include the use of wavelet packets, again in a rate-distortion optimizing framework [65]. They found that packets can yield superior performance to a pyramid decomposition with certain types of images, including an FBI fingerprint image. The packet transform used is a bottom-up approach in which *all* sub-bands are decomposed to a certain level and then branches of the quadtree can be pruned to yield the minimal distortion for a given bit-rate [48]. Bottom-up techniques can be computationally intense, since the wavelet transform must be performed until every sub-band reaches a certain resolution. The rate-distortion optimization can require a high degree of computation as well.

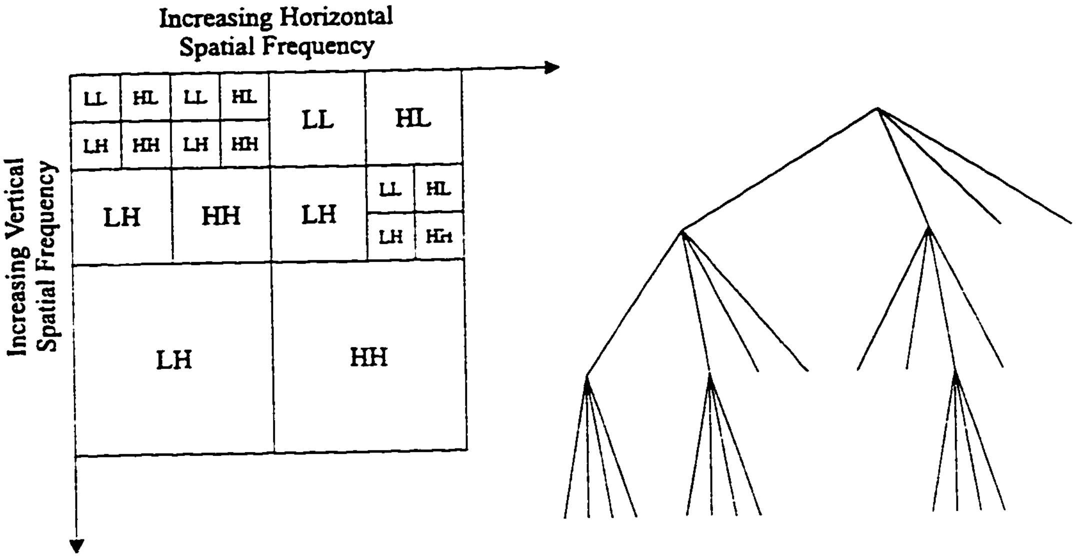

Increasing Horizontal Spatial Frequency
Increasing Vertical Spatial Frequency
LL
HL
LL
HL
LH
HH
LH
HH
LL
HL
LH
HH
LH
LL
HL
LH
HH
LH
HH

Figure 2.5: A 3-level wavelet packet decomposition and its quadtree representation.

Chang and Kuo used wavelet packets to analyze and classify various textures [10]. The multi-resolution wavelet transform is excellent for distinguishing textures of different scales. Packets add the benefit of allowing further decomposition of specific band-pass frequencies which is effective because "a large class of natural textures can be modeled as quasi-periodic signals whose dominant frequencies are located in the middle frequency channels" [10]. Their technique involves a top-down decomposition, in which the decision to decompose a sub-band is based on a calculation of its energy.

We will be delving further into wavelet packets in subsequent chapters. The wavelet packet decomposition may be more flexible than a pyramidal technique in taking advantage of the human visual system, since visually significant sub-bands can be further decomposed.

2.5 Quantization

In order to be digitally processed, the original continuous signal of the image must have its amplitude and spatial extent digitized into discrete values. Amplitude digitization is known as intensity quantization and produces the digital image $f(x, y)$ [28].

Digital images can be further quantized during compression. As mentioned in

Chapter 1, images can require large amounts of storage space. For example, a 512 × 512 image with the standard 2^8 intensities requires 256 kilobytes of memory. The same image quantized to 2^6 intensities would use 192 kilobytes. Note that by simply quantizing the intensities we can already achieve some compression, but by using a quantizer as part of an image compression system we can achieve far superior results.

The image resulting from quantization will no longer be equal to the input image, i.e., some unrecoverable errors will be introduced. An image can be quantized to a certain precision before the errors are perceptable, but if the quantization is too coarse (i.e., not enough distinct intensities are used), the reconstructed image may exhibit false contours, which can be visually annoying.

Quantization is simply a form of "rounding off" and therefore can be applied to whatever form the signal information is in after it has passed through the mapper, be it predictive coding errors or transform coefficients.

There are several forms of quantization used in image compression, ranging from simple uniform scalar quantization to vector quantization.

2.5.1 Uniform Scalar Quantization

The process of quantization rounds off values in the range between z_k and z_{k+1} to output value q_k. (This range is often called a quantization bin.) Given a probability distribution of values to be quantized, we can choose to make all quantization bins equal-sized. This quantization strategy is called uniform quantization and is depicted in Figure 2.6.

The technique is very popular due to its computational simplicity and decent performance. Uniform quantization is used in some of the top-performing methods reported in the literature, including the embedded zerotree wavelet algorithm (EZW) [53], the space-frequency quantization algorithm (SFQ) [64], set partitioning in hierarchical trees (SPIHT) [51], the FBI fingerprint image compression standard [6], and in the standard JPEG scheme [57].

One drawback of this method, however, is that the same quantization error will be introduced on higher probability values as on lower probability values (unless the input probability distribution is uniform, which is highly unlikely after passing through the mapper). This fact has lead to variable-bin-size, non-uniform quantization.

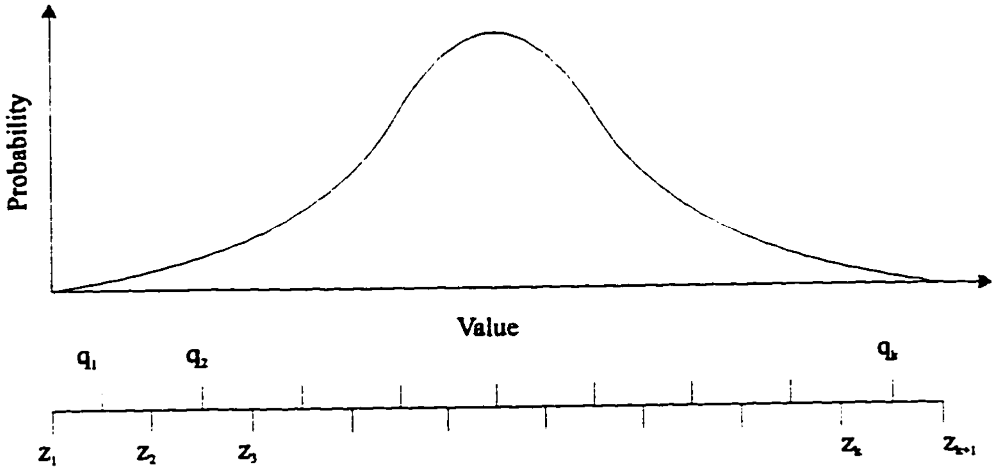


Figure 2.6: Uniform scalar quantization given an input probability distribution.

2.5.2 Non-Uniform Scalar Quantization

In order to minimize the mean-square error resulting from the process of quantization. the bin sizes should be optimally calculated based on the probability distribution of the input values as shown in Figure 2.7.

For a quantizer of K' output levels, the bin boundaries can be calculated as

$$z_k = \frac{q_{k-1} + q_k}{2} \quad \text{for} \quad k = 2, 3, \ldots, K', \tag{2.11}$$

and the output values are then

$$q_k = \frac{\int_{z_k}^{z_{k+1}} zp(z)\,dz}{\int_{z_k}^{z_{k+1}} p(z)\,dz} \quad \text{for} \quad k = 1, 2, \ldots, K', \tag{2.12}$$

where $p(z)$ is the probability density function of the input value z.[5] Note that the optimal non-uniform quantizer will have its bin limits halfway between each output level, and its output levels will be in the centroid of each bin.

It is difficult to find a solution that simultaneously satisfies both equations (2.11) and (2.12). Given z_1 and z_K, one technique involves estimating q_1 and calculating the remaining values using equations (2.12) and (2.11). This trial-and-error method will

[5] For a complete derivation of these formulae see [49].

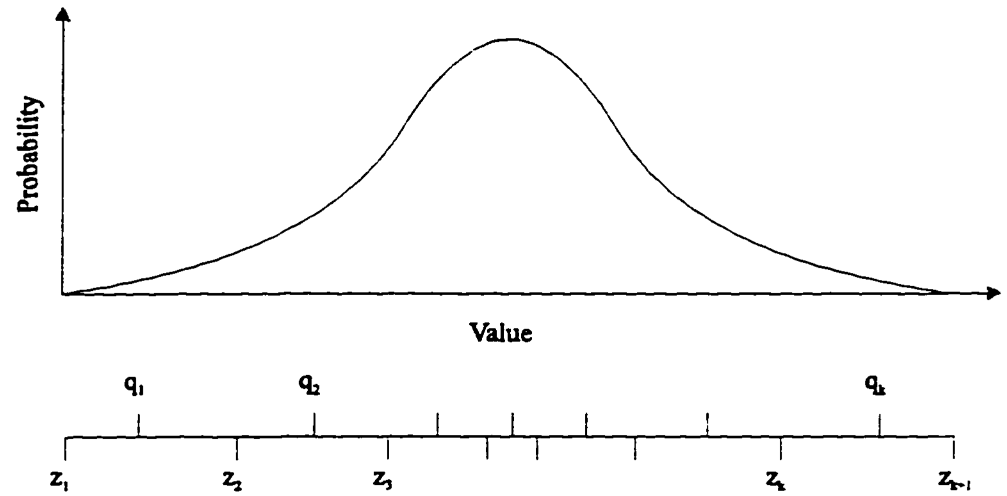


Figure 2.7: Non-uniform scalar quantization given an input probability distribution.

be successful if the final output level lands in the centroid of the last bin. Max used this technique to find the optimal non-uniform quantizer for an input with a Gaussian distribution [49].

In order to avoid this complicated solution, one can use the technique called *companding* [26]. Using companding, the input values are first mapped to new values by a *compressor* function $G(x)$. This function normally has large slope for small magnitudes and small slope for large magnitudes, which results in "spreading out" the distribution of the input values. Once this is accomplished, a simple uniform quantizer can be used. During decompression, an *expandor* is used, namely $G^{-1}(x)$, to recreate the rounded off original values. Logarithmic companders are often used because

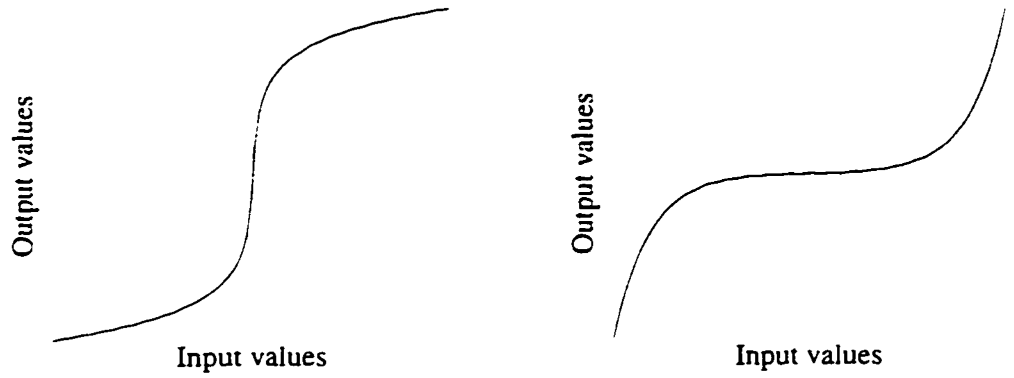
> *for low signal power levels, most samples are small in magnitude and therefore small step sizes are desirable for an adequate SNR. For high signal levels, a large fraction of the samples have very large magnitude and a larger step size can be used to maintain a given SNR objective.* [26]

A typical compressor and expandor function pair used in speech coding is the μ-law characteristic,

$$G_\mu(x) \;=\; V\frac{\ln(1+\mu|x|/V)}{\ln(1+\mu)}\mathrm{sgn}(x) \quad \text{for} \quad |x| \le V, \tag{2.13}$$

$$G_\mu^{-1}(x) \quad = \quad V\frac{(\exp(\ln(1+\mu)|x|/V) - 1)}{\mu}\mathrm{sgn}(x) \quad \text{for} \quad |x| \leq V, \qquad (2.14)$$

where μ controls the amount of companding. An example $G_\mu(x)$ and the corresponding $G_\mu^{-1}(x)$ are shown in Figure 2.8.



(a) $G_\mu(x)$

(b) $G_\mu^{-1}(x)$

Figure 2.8: Example (a) compressor and (b) expandor functions for companding with $\mu = 255$.

2.5.3 Vector Quantization

The quantizer can be built using a different technique called vector quantization (VQ), in which *vectors*, or blocks of values are rounded off together. For detailed analysis of different VQ techniques. see [29, 44, 26, 1, 13]. The process can be described with the following steps.

1. **Extract vectors from training images.** Normally the *training* images should be a representative sample of the images one plans to compress.

2. **Generate a VQ codebook.** The *codebook* can be constructed from the training vectors by choosing the most representative vectors from the training images. This step can often be accomplished through *clustering* techniques to minimize the distortion. The codebook must be stored at the encoder and decoder and usually contains from 2^6 to 2^9 vectors of size 2×2 to 8×8.

3. **Quantize image.** Extract vectors from the input image. Quantize these vectors to the "closest" (e.g., in a mean-square error sense) codevectors in the codebook.

4. **Transmit vector indices.** Send the codebook indices of the codevectors through the channel. This step achieves compression since the vectors themselves need not be sent.

5. **Reconstruct image.** The decoder simply executes a table look-up on the codevector indices and rebuilds the image from the vectors.

The VQ encoding (step 3 above) and decoding (step 5 above) steps are shown in Figure 2.9, which is adapted from [18]. Note that the reproduced vector may not be identical to the input vector.

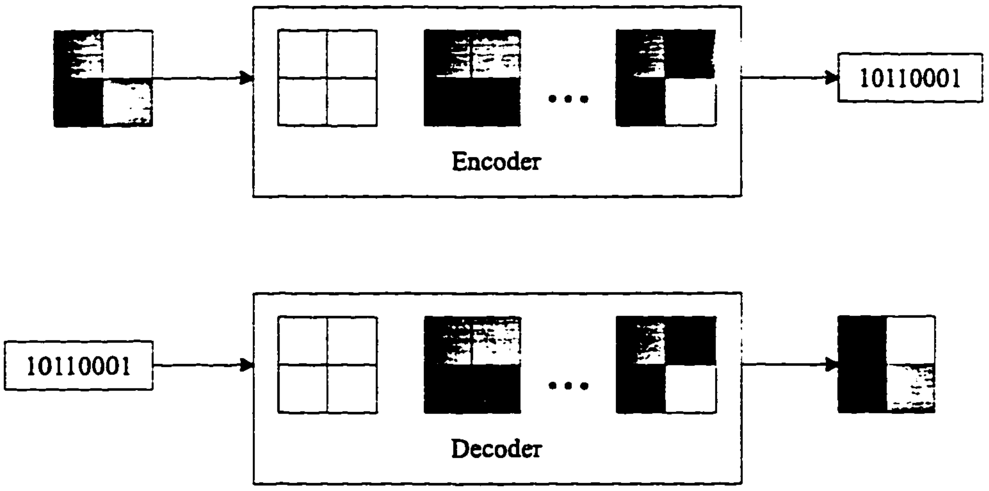


Figure 2.9: Vector quantization.

The first VQ codebook generation procedure was presented by Linde, Buzo, and Gray and is known as the LBG algorithm [37]. The algorithm is described as follows.

1. **Initialize codebook.** The codevectors could be populated by random sampling vectors from training images.

2. **Cluster training vectors.** Label each training vector as belonging to a cluster associated with its nearest neighbor codevector, as calculated by a suitable distortion measure.

3. **Recalculate codevectors.** Manipulate each codevector in the codebook in order to minimize the average distortion of all the training vectors in its cluster (i.e., find the centroid of the cluster).

4. **Iteration.** Repeat steps 2 and 3 until the change in the codebook is small. The number of iterations of this algorithm, therefore, cannot be predicted.

Despite the fact that VQ has been shown to be theoretically superior to scalar quantization [28], there are several practical difficulties associated with the technique [43], including:

- high computational complexity (at the encoder) which grows exponentially with vector dimension;

- sensitivity of the codebook to training images, i.e., it is difficult to get a representative sample, especially of edges;

- noticeable errors in the reconstructed image (e.g., false contours) due to the limited size of the codebook:

- and edge degradation, especially when VQ is applied directly to the image pixels. The reconstructed image can exhibit a staircase-like effect since it is built from VQ blocks.

Many researchers have attempted to address the above problems. Several different techniques have been suggested to reduce the complexity of the codebook generation process [22, 30, 1, 9]. In most cases, the new methods can decrease the complexity substantially from the LBG algorithm without significantly increasing distortion. Lattice VQ techniques can produce VQ codebooks algebraicly and do not require codebook storage or complex procedures for its generation [52], but codebooks designed specifically for a certain type of image will always perform better than lattices [1]. Generally the codebook generation procedure only needs to be executed once, however, so reducing the generation time may not be significant.

One way to attack the edge degradation problem is to use more than one codebook. *classified* into different types of vectors. Mohamed and Fahmy [43] treat high-detail and low-detail portions of the image separately by designing multiple codebooks. Another way to deal with the limited size of the codebook is to use *adaptive* techniques in which the codebook can change over time by introducing new codevectors and sending them to the decoder [27].

The VQ encoding process can also be time consuming because a best-fit codevector must be found for each input vector. Rather than using a linear search technique on a codebook, a tree-structured VQ (TSVQ) could be used [18]. The TSVQ codebook is often a binary tree with a codevector placed at each node. In this case the search begins at the root of the tree. The encoder compares the distortion of the current node with the children and follows the path with the minimum distortion. This procedure continues until a leaf node is hit and is essentially a binary search. The sequence of binary decisions is sent to the decoder as an index so that it can look up the proper codevector. The storage requirement is increased over other codebooks since leaf nodes are the only true codevectors, but the remainder of the tree containing the test vectors must also be stored.

Work has been done in using VQ in conjunction with second generation methods [55] and with characteristics of the HVS [8] by using filter preprocessing of color images to model the impulse response of the eye.

Many researchers (and image compression standards, including the FBI fingerprint and JPEG specifications) have found that in practical applications, the more complicated methods of non-uniform or vector quantization do not produce improvements in image quality over simple uniform quantization [57, 5, 2, 64, 65]. The good performance of uniform quantizers may be due to the fact that an optimally designed uniform quantizer coupled with an entropy coder will perform similarly to a non-uniform quantizer [28]. We will be looking into the trade-offs of using a non-uniform quantizer in more detail, specifically examining its efficacy in modeling a portion of the human visual system.

2.6 Image Quality Assessment

Once the image has been quantized and reconstructed for viewing, the quality should be measured in order to assess the performance of the image compression scheme. The most popular measure used to assess the quality of the reconstructed image is the peak signal-to-noise ratio (PSNR),

$$\mathrm{PSNR} = 10 \log_{10} \frac{P^2}{\mathrm{MSE}}, \tag{2.15}$$

where P is the peak signal (usually 255 for a grayscale image) and MSE is the mean-square error. The MSE for an $M \times N$ image $f(x, y)$ and its reconstruction $\hat{f}(x, y)$ is

$$\mathrm{MSE} = \frac{1}{MN} \sum_{j=1}^{M} \sum_{k=1}^{N} \left(f(j, k) - \hat{f}(j, k) \right)^2 . \tag{2.16}$$

PSNR gives an objective mathematical measure of image quality, but is well-known to poorly correlate with subjective human evaluation of quality [26, 56, 23, 13, 31].

Since human observers are the final recipients of the reconstructed image. some experiments involve the use of large groups of expert and non-expert users to judge image quality by rating the image on a given distortion scale (e.g., 1 = not noticeable. 2 = just noticeable, ..., 7 = extremely objectionable) [23]. These experiments are costly and time-consuming. Perhaps a better approach would be to incorporate known characteristics of the human visual system (HVS) into the development of a quantitative image quality measure.

2.7 Relevant Aspects of the Human Visual System

The human visual system (HVS) is mainly sensitive to changes in three qualities: luminance or light level, spatial frequency, and signal content [19].[6] An important point is that modeling the HVS requires separation of these sensitivities, but their interactions can be complex.

2.7.1 Luminance Sensitivity

It has been shown that the eye is non-linearly sensitive to changes in amplitude, due to the retina adapting to ambient light changes [13]. The eye is much more capable of distinguishing intensity variations in a dark area,[7] which is due to certain physiological characteristics of the eye [54].

Light is transmitted through the iris onto the retina, where it meets two types of photoreceptors, *rods* and *cones*. Rods are very sensitive to luminance and distinguish shades of gray. The cones see color and provide greater discrimination of details. Cones quickly lose their ability to sense any signal with decreasing brightness. Therefore, in low-light levels, the cones provide minimal color or detail discrimination.

2.7.2 Spatial Frequency Sensitivity

Experiments have shown that the HVS responds like a Fourier analyzer to a first approximation [54]. Assuming this statement is true, we can find out what frequencies the eye is sensitive to. To determine the frequency sensitivity, experiments have been conducted by displaying sine waves of varying spatial frequencies (and, therefore. of varying contrast) on a monitor and having a subject indicate whether or not the

[6] The HVS also exhibits variable sensitivity to color and motion, but these do not apply to this research.

[7] For example, we can see the stars at night but not during the day [4].

cycles of the wave can be discriminated. The spatial frequency is measured in cycles per degree, which is the number of wavelengths of a sine wave in terms of an angle subtended from the eye (see Figure 2.10).

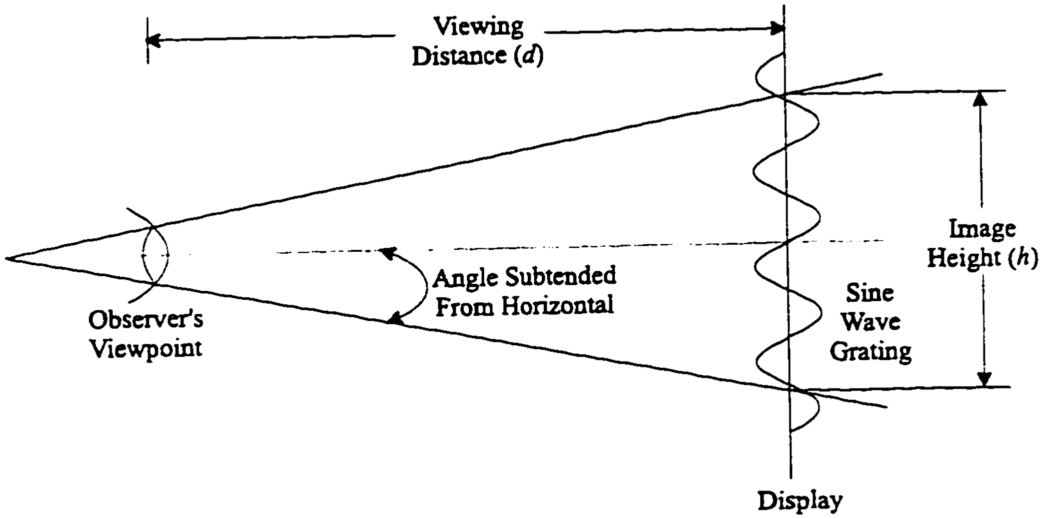


Figure 2.10: Spatial frequency sensitivity experiment setup.

The visibility threshold can then be plotted as a function of spatial frequency, termed the contrast sensitivity function (CSF). Any signal of frequency below this threshold will not be visible. It has been found that the eye has a band-pass frequency response peaked around 4–8 cycles per degree. The orientation of the image signal is visually important as well. The spatial frequency sensitivity is *anisotropic*, that is, humans are more sensitive to horizontally- and vertically-aligned signals than to those in the diagonal [46]. Figure 2.11 shows one quadrant of a typical CSF [42]. Note that the sensitivity is normalized.

2.7.3 Signal Content Sensitivity

The actual content of the image also affects the sensitivity of the HVS. In general, portions of an image which are highly active with detail can be degraded with larger amounts of distortion (with the restriction that the noise be of the same spatial-frequency and orientation of the signal) than smoother regions before this distortion is visible. This process is referred to as *masking* since high-pass signals can "mask" the distortion. The existence of masking actually suggests that the accuracy of the

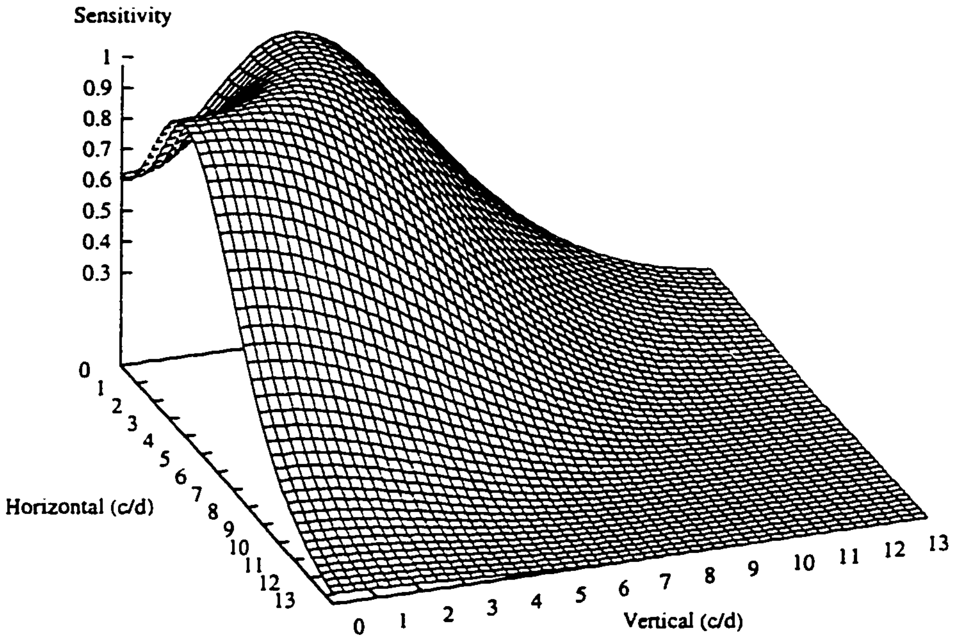


Figure 2.11: Example anisotropic contrast sensitivity function.

contrast sensitivity function does not always hold while in the presence of a non-uniform signal. This should not come as a surprise, however, since the CSF was arrived at through experiments with uniform sine gratings.

In general, the masking effect increases with signal intensity, but the specific shape of this function in real images is not known [19]. Another complication is the effect of *facilitation* [32]. Constructive interference between the content and noise signals can actually facilitate, i.e., increase, the visibility of distortion. This effect can make it difficult to take advantage of the masking effect in a codec.

2.7.4 The Human Visual System and Quality Assessment

Understanding some qualities of the HVS should aid in developing superior compression systems and quality measures. People have only begun to understand and incorporate HVS characteristics into compression schemes and have had fairly limited success thus far [13].

There have been many attempts to develop a quality measure which correlates well to human perception [40, 60]. Xu and Hauske develop a measure by identifying several types of image distortion: noise in smooth areas, artificial blocking structures, false contouring, edge blurring, and ringing [66]. Miyahara *et al.* develop a picture quality scale (PQS) with a similar technique based on several error factors, including random errors, end-of-block disturbances, correlated errors, and edge errors [42]. The factors are then weighted and joined in a manner which results in a high correlation with observers' opinions. PQS was used to judge the reconstructed image quality from various wavelet-based codecs by separately examining the steps of wavelet transform, quantization, and entropy coding [39]. Kim *et al.* developed a weighted mean-square error taking into account human sensitivity to different spatial frequencies [31].

The idea that one number should be capable of expressing the quality of an image may be unrealistic, considering the variables of display monitor, human viewer, ambient lighting, and image content. Daly developed a visual difference predictor (VDP) which produces a "map" image indicating the location and detection probability of visible image distortion [19]. A system like this would allow more focused research into what qualities of a quantizer are causing the most visible errors.

2.7.5 The Human Visual System and Image Compression

Some researchers have attempted to use a weighting system, in which perceptually important areas are weighted highly and are therefore finely quantized. As previously discussed, second generation techniques attempt to segment the image into perceptually important and unimportant regions in the spatial domain. Each region can then be weighted and quantized accordingly.

Experiments by psychophysicists and visual psychologists have determined that the HVS "filters the image into a number of bands ... [and that] the image should be considered to be composed of information at a number of different scales." [36]. This observation suggests that the wavelet transform already models the HVS in its manner of decomposing an image, which inspired Kim *et al.* to use HVS weighting in conjunction with wavelet coefficients [31].

O'Rourke and Stevenson also use the wavelet transform and then weight the subbands according to orientation and contrast sensitivity of the HVS [46]. Using a given desired bit-rate, an adaptive VQ scheme is applied to efficiently allocate bits to the coefficients, allotting larger codebooks for highly-weighted sub-bands. The computational complexity is high since a new VQ codebook must be designed and transmitted for each sub-band. Their method is found to work better than JPEG on smooth regions, but exhibits more aliasing at edges. The aliasing may be due to one of the problems with VQ already mentioned—significant end-of-block errors can occur. In addition, this horizontally- and vertically-aligned distortion can be annoying due to the orientation sensitivity of the HVS.

Watson *et al.* have done controlled experiments to determine a visible wavelet quantization error threshold for each frequency sub-band in the presence of a uniform background signal [58, 59]. Lewis and Knowles design their uniform wavelet coefficient quantizers by attempting to model the effects of "the background luminance, the proximity to an edge, the frequency band, and texture masking." [36].

2.8 Summary

This chapter has presented an overview of image compression and has covered the fundamentals of general image coding systems. This discussion has included techniques which have been used to reduce the redundancies found in images. Mapping techniques including predictive coding, transform coding (stressing the wavelet techniques), and second generation approaches were discussed. To achieve significant compression, the mapped values must be quantized by some variation of uniform, non-uniform, or vector quantization. Understanding some characteristics of the human visual system can increase our ability to focus on important aspects of the image compression process and also to develop more accurate image quality measures.

The stage is now set to determine the scope of the the work done in this thesis and the justification for those boundaries. The following chapter details the methodology and areas explored.

Chapter 3

Design of the WP/HVS Framework

We have seen in the previous chapter a subset of the myriad approaches to the image compression problem. Proponents of second generation techniques suggest that the future of image compression may lie in image understanding and modeling. However, second generation methods face a large number of challenges, as discussed in Section 2.3.2. Wavelet-based compression is less computationally intense than most second generation schemes and has been able to achieve high compression rates (e.g., 20:1) while maintaining good image quality. Most researchers do agree that aspects of the human visual system (HVS) need to be incorporated into compression, but how should this be done, and what effects might it have?

We propose an image compression framework incorporating HVS characteristics in conjunction with a flexible, computationally-light, wavelet packet decomposition. This framework will be referred to as the WP/HVS framework, since *w*avelet *p*ackets and *HVS* modeling are core components. This chapter discusses that framework in detail by providing:

- a high-level blueprint of the framework;
- justification for the choice of each component in the framework;
- and in-depth design and implementation details for each component.[1]

[1] Some fundamentals of the basic components (e.g., uniform quantizer, wavelet transform, and so on) will be left out, since this information can be found in Chapter 2.

3.1 Blueprint of the Framework

This section presents a brief overview of the proposed WP/HVS framework and the processes carried out within it. Figure 3.1 is a detailed block diagram of the framework.

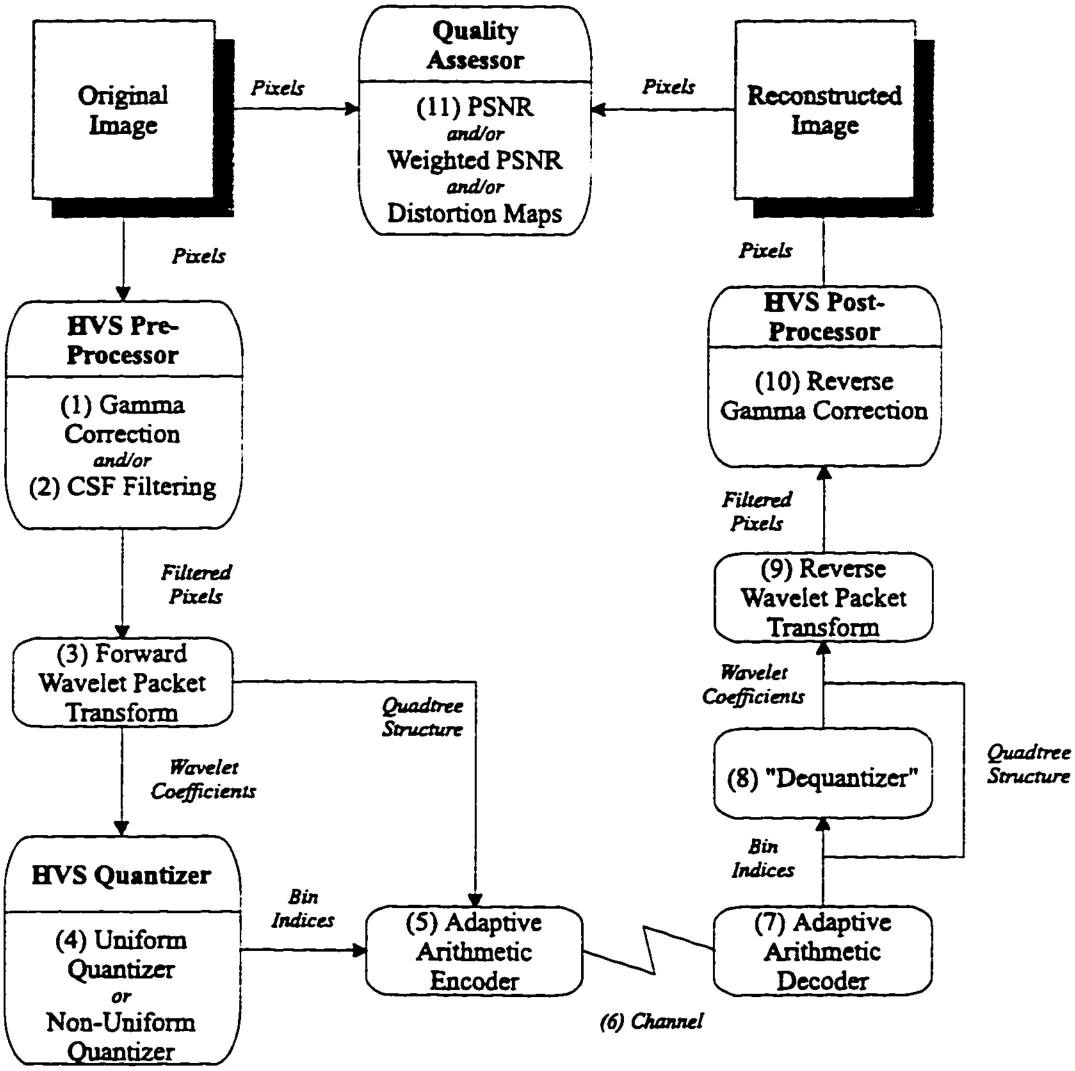


Figure 3.1: Block diagram of the WP/HVS framework.

The diagram depicts a series of steps executed throughout the WP/HVS framework. In the following list, the numbered steps correspond to the numbers in brackets in Figure 3.1.

1. **Forward gamma correction.** Optionally gamma correct the pixels of the input image to model the luminance sensitivity of the HVS.

2. **CSF filtering.** Optionally filter from the original image the frequencies to which the HVS is insensitive, as represented by the contrast sensitivity function (CSF). This step is *lossy* since the filtering permanently removes some frequencies. However, this process is designed to be *visually lossless* in that the output image should not be visually different from the original.

3. **Wavelet packet decomposition.** Perform the forward wavelet packet transform using a top-down decomposition approach. The quadtree (QT) representation of the wavelet sub-band structure is entropy coded separately.

4. **Wavelet coefficient quantization.** This step can be accomplished using uniform or non-uniform quantization, both being designed with HVS sensitivity in mind. A quantizer with unique parameters is designed for each wavelet packet sub-band. The output from the quantizer is the bin index calculated for each coefficient.

5. **Coefficient symbol encoding.** Entropy code quantization bin indices with an adaptive arithmetic coder. The coder can adapt to the source statistics of the quantized coefficients in each sub-band. This step signals the end of the compression process.

6. **Channel.** The encoded information can be stored on disk or transmitted across a network.

7. **Coefficient symbol decoding.** Decompression begins by using the adaptive arithmetic decoder to retrieve the quantization bin index of each wavelet coefficient.

8. **Wavelet coefficient "dequantization".** Once quantization has occurred, it is not possible to retrieve the lost information, but each rounded wavelet coefficient must be calculated from its bin index.

9. **Wavelet packet reconstruction.** The reverse wavelet packet transform is performed on the coefficients to return to the image domain defined by pixel intensities. The exact order of sub-band reconstruction is dictated by the QT structure.

10. **Reverse gamma correction.** If gamma correction was performed on the original image, the reverse process must now be executed.

11. **Quality assessment.** At this point we are left with the reconstructed image. The quality assessor can be used to judge the reconstructed image in comparison with the original using the peak signal-to-noise ratio (PSNR), a PSNR weighted by the frequency sensitivity of the HVS (WPSNR), or distortion maps showing the location of information loss.

The remainder of this chapter will focus on each *significant* framework component, explaining its inner workings and the justification behind its use. We will concentrate on the HVS pre-processing steps which model the luminance and frequency sensitivity, the wavelet packet decomposition, the quantizers, and the quality assessment tools.

3.2 Modeling Luminance Sensitivity

Distributing the available gray levels evenly over the range from black to white ignores the fact that the HVS is sensitive to ratios of intensity as opposed to absolute differences. For example, we perceive the difference in brightness between gray levels 10 and 11 to be the same as that between 50 and 55 [25]. The amplitude non-linearity of the HVS can be compensated for by a "gamma correction" pre-processing step, in which the gray levels are assigned approximately logarithmically rather than linearly [25]. Each input pixel $f(i,j)$ is changed to a new value,

$$g(i,j) = kf(i,j)^{\frac{1}{\gamma}}, \tag{3.1}$$

where k is a factor used to hold constant the dynamic range of the image. If m is the maximum value of $f(x,y)$ then

$$k = \frac{m}{m^{\frac{1}{\gamma}}}. \tag{3.2}$$

The range of γ is normally $2.2 \leq \gamma \leq 2.5$ [25]. Since most computer display hardware is manufactured in a manner that compensates for the HVS luminance sensitivity, we need to reverse the process in the image reconstruction phase. Otherwise, the image will appear to be *twice* gamma corrected.[2] Figure 3.2 presents example graphs for forward and reverse gamma correction functions using $\gamma = 2.2$. Implementing the gamma correction as a pre-processing step provides the flexibility to assess compression results with or without it.

[2] However, applying gamma correction in software would likely be required only when it is not supported in hardware. We shall examine this aspect in more detail in Chapter 4.

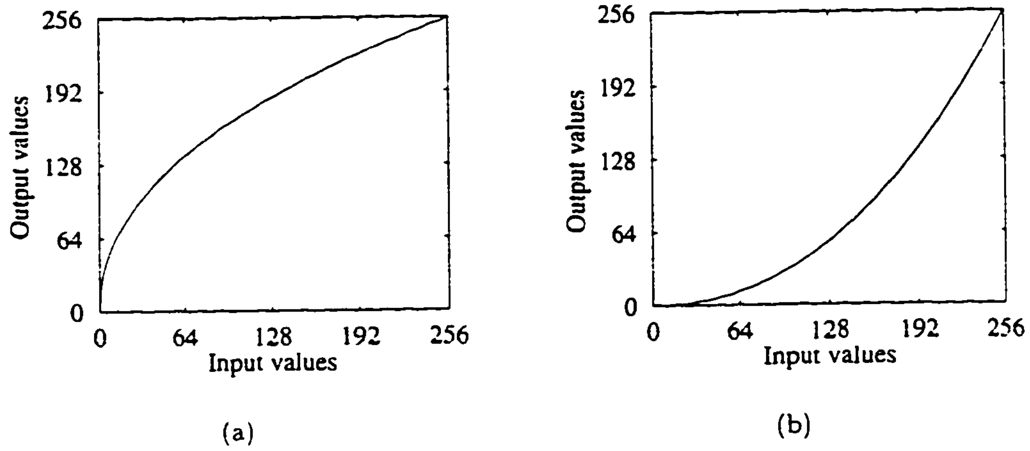


Figure 3.2: Example (a) forward and (b) reverse gamma correction with $\gamma = 2.2$.

3.3 Modeling Frequency Sensitivity

Recall from Section 2.7.2 that experiments have shown that the HVS has varying sensitivity to different spatial frequencies, as modeled by the contrast sensitivity function (CSF). We have chosen to implement a pre-processing filter to model this HVS characteristic using the fast Fourier transform (FFT). Similar to the gamma correction option, this modular approach will facilitate experimentation using different HVS characteristics. It will also make it easier to determine what portions of the framework have what effects on the reconstructed image. Most researchers do not report on the same codec with and without modeling certain HVS effects [36, 46, 31]. The WP/HVS framework has been designed to support such research.

Another reason for not embedding the CSF filter into the mapper is the difficulty associated with accurately representing it in the wavelet domain. The Fourier domain directly represents image signals with sine and cosine basis functions, which is exactly the type of experimental data used in determining the shape of the CSF.

Please refer to Figure 2.11 on page 32 showing one CSF model for the HVS frequency sensitivity.[3] How was this model designed? Let us look in more detail at how to model the HVS frequency sensitivity.

[3] The CSF model used here is simply *one* model. Other models have been developed by other researchers [54, 19].

3.3.1 Modeling an Isotropic CSF

According to the model used by Miyahara *et al.* [42], the isotropic spatial frequency sensitivity $\mathrm{CSF}_i(\omega)$ can be approximated by

$$\mathrm{CSF}_i(\omega) = 1.5\exp(-\sigma^2\omega^2/2) - \exp(-2\sigma^2\omega^2), \tag{3.3}$$

where $\sigma = 2$ and ω is the angular expression of the frequency. This conversion is defined as

$$\omega = \frac{2\pi f}{60}, \tag{3.4}$$

where

$$f = \sqrt{u^2 + v^2}, \tag{3.5}$$

in which u and v are respectively the horizontal and vertical spatial frequencies in the Fourier domain.

Finding the spatial frequencies from the Fourier transform, given a viewing distance from the image, is a fairly simple process. As an example, consider the problem of calculating the maximum spatial frequency in one dimension given a viewing distance d stated in terms of multiples of image height h. The viewing angle δ (see Figure 2.10 on page 31) can be found by

$$\delta = \tan^{-1}\left(\frac{h/2}{hd}\right) = \tan^{-1}\left(\frac{1}{2d}\right). \tag{3.6}$$

The number of *pixels* per degree of the viewing angle δ can then be calculated with

$$\text{pixels/degree} = \frac{h}{2\theta}, \tag{3.7}$$

where h is the image height in pixels. This quantity is actually the sampling rate for the displayed image. Signal processing theory states that any signal can be reconstructed accurately as long as it is sampled at greater than the Nyquist rate [25]. The Nyquist rate is twice the highest possible spatial frequency, which yields the simple relation

$$\text{cycles/degree} = \frac{\text{pixels/degree}}{2}. \tag{3.8}$$

Figure 3.3 shows the resulting CSF based on the model of equation (3.3). Recall from Section 2.7.2 that the orientation of the image signal is also visually important, since humans are less sensitive to signals on a diagonal. Miyahara *et al.* show how to extend the CSF model to take this orientation sensitivity into consideration [42].

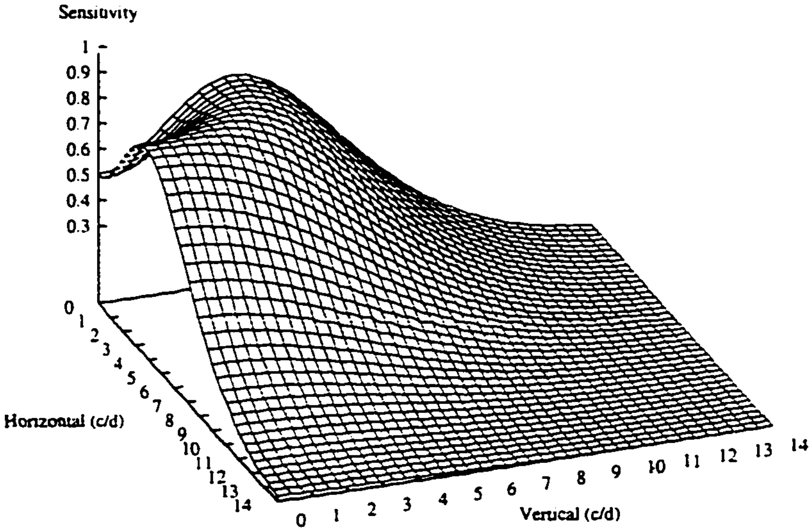


Figure 3.3: Surface plot of a quadrant of the isotropic CSF model CSF_i, viewed at a distance of three times the picture height.

3.3.2 Extending the Model to an Anisotropic CSF

The HVS spatial frequency sensitivity is *anisotropic*, that is, humans are more sensitive to horizontally- and vertically-aligned signals than to those in the diagonal [46]. The anisotropy is mainly present in higher-frequencies beginning at about $f_a = 11$ cycles per degree. This effect can be modeled by a filter of the form

$$\mathrm{CSF}_a(\omega, \phi) = \frac{1 + \exp\left(\beta(\omega - \omega_a)\right)\cos^4 2\phi}{1 + \exp\left(\beta(\omega - \omega_a)\right)}, \tag{3.9}$$

where $\beta = 8$ and ϕ is the angle between the horizontal and vertical frequency components,

$$\phi = \tan^{-1}\left(\frac{v}{u}\right). \tag{3.10}$$

Figure 3.4 depicts $\mathrm{CSF}_a(\omega, \phi)$. Notice that this filter has unit value for all spatial frequencies $f < f_a$. This band-pass response facilitates forming the overall anisotropic CSF model through multiplying by the isotropic model,

$$\mathrm{CSF}(u, v) = \mathrm{CSF}_i(\omega) \times \mathrm{CSF}_a(\omega, \phi). \tag{3.11}$$

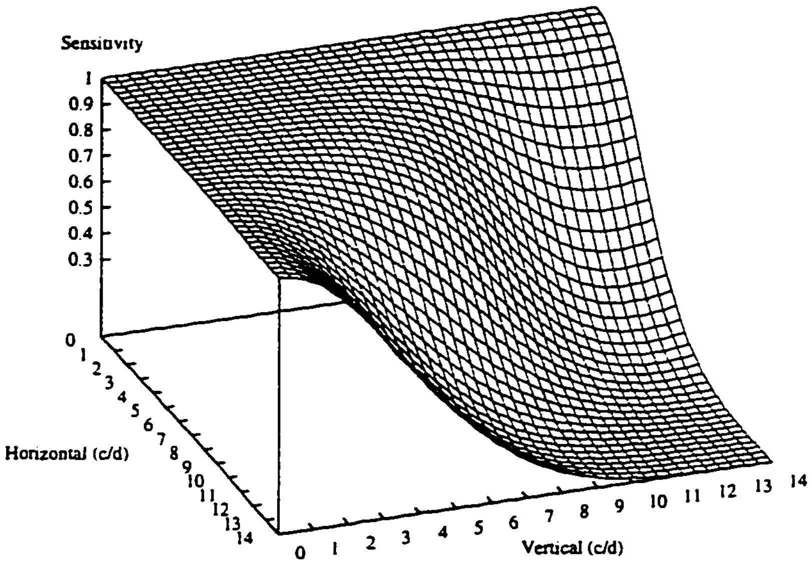


Figure 3.4: Surface plot of a quadrant of the anisotropic CSF model CSF_a, viewed at a distance of three times the picture height.

The result of the model of equation (3.11) was already presented in Figure 2.11 on page 32. and now its origin should be clear. However, we still have not explained how the CSF is applied in the HVS pre-processing filter operation.

3.3.3 Applying the CSF Model in the WP/HVS Framework

The CSF model is implemented in the framework by an FFT filtering operation. We have chosen not to directly use $\mathrm{CSF}(u, v)$ for designing the filter, since this model assumes that the observer is viewing the image at one exact distance from the monitor. which may be unrealistic. We can allow more flexibility by accepting a range of viewing distances [19].

Allowing the viewing distance to be set within a range suggests that we have two CSF filters, one for the near viewing distance d_n, and one for the far viewing distance d_f, again expressed in terms of the picture height h. The overall CSF can be defined as the envelope of the two filters which will yield a band-pass filter like that in Figure 3.5 (shown using $d_n = 2h$ and $d_f = 4h$). Note the flat portion at the peak of sensitivity,

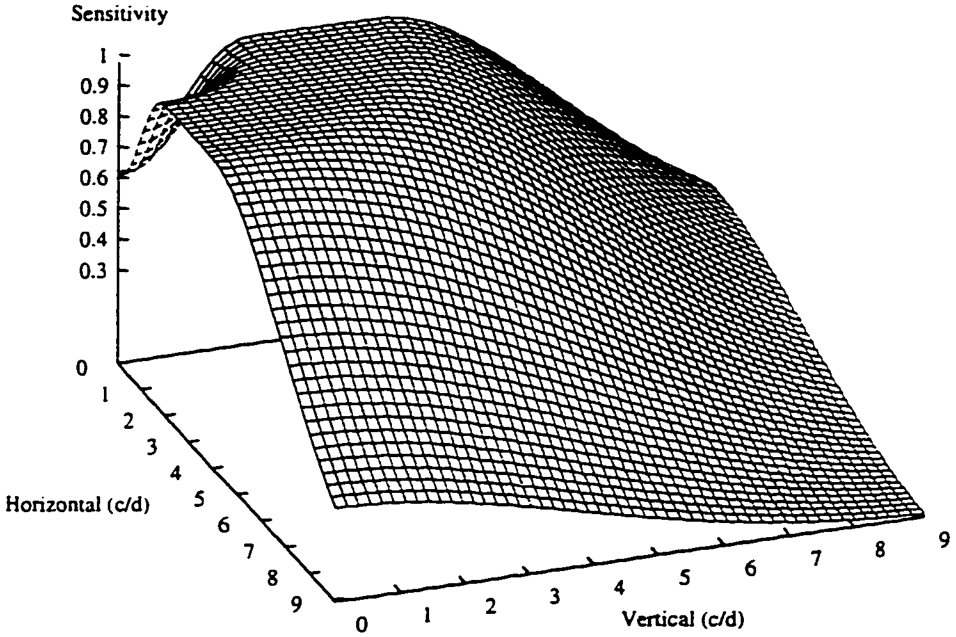


Figure 3.5: Example surface plot of the CSF filter used in the WP/HVS framework with near viewing distance $2h$ and far viewing distance $4h$.

which is where the envelope of the two CSF filters is formed. The power of the spatial frequencies in this area will not be modified by our CSF filter.

Once the image has passed through the HVS pre-processor, which deals with the luminance and frequency sensitivity, the result is ready to be transformed by the mapper, using a wavelet packet decomposition.

3.4 Wavelet Packets

The mapping architecture we have chosen to use is a wavelet packet transform. We have avoided second generation approaches due to the problems associated with segmentation, modeling, and structure overhead. One of the main thrusts behind second generation techniques is that they attempt to compress the image in an HVS-relevant manner, by segmenting the image into its component parts in the spatial domain. However, using wavelet decomposition can efficiently mimic the HVS in the transform domain. Experiments have found that the human eye actually filters image data into several frequency sub-bands and treats the image as if it were composed of information at many different resolutions [36]. In addition, recall from Section 2.4.2 that wavelet compression facilitates simple progressive transmission, whereas it is unclear how to accomplish such transmission with second generation techniques.

Typical wavelet packet use involves full decomposition of all sub-bands, followed by back-tracking to a point where a rate-distortion criteria is met. This may yield an optimal packet decomposition (in a rate-distortion sense) but incurs a great deal of computational overhead. We attempt to simplify the process by using a recursive decomposition with a stopping criterion.

Wavelet packets are adaptive to the signal content of the image, since the decision to decompose a sub-band is made dynamically, unlike with the standard forced pyramidal decomposition. Using wavelet packets also offers more flexibility for incorporating HVS characteristics since the significant frequency sub-bands are clearly defined. However, we will also explore the traditional pyramid decomposition since it is a special case of the packet transform.

This section describes the functionality of the wavelet packets, including the decomposition algorithm, the options for determining when a given sub-band should not be decomposed, and a description of the quadtree (QT) structure.

3.4.1 Decomposition Algorithm

What follows is a detailed representation of the wavelet packet decomposition algorithm. Let $S(\lambda,\theta)$ be the sub-band at decomposition level λ (where $\lambda = 1,\ldots,\lambda_{\max}$) orientation θ (where $\theta = 1,\ldots,4$). The original image is then denoted $S(0,0)$. The stopping criterion calculated on a sub-band is $C(S(\lambda,\theta))$. Note that the maximum depth of decomposition is also set and is denoted $\lambda_{\max}$.

1. **Initialize.** Find $c_0 = C(S(0,0))$. Create empty QT. Send c_0 and $S(0,0)$ to step 2.

2. **Decompose.** Store a copy of the given sub-band, which will be called $S(\lambda,\theta)$. Convolve the rows and then the columns with the low- and high-pass subsampling wavelet filters L and H,

$$\begin{aligned} S(\lambda,\theta) &= L * \text{rows}(S(\lambda,\theta)) + H * \text{rows}(S(\lambda,\theta)), \\ S(\lambda,\theta) &= L * \text{columns}(S(\lambda,\theta)) + H * \text{columns}(S(\lambda,\theta)). \end{aligned}$$

   Split the sub-band $S(\lambda,\theta)$ into four quadrants $S(\lambda+1,t)$, where $t = 1,\ldots,4$.

3. **Stopping criterion.** Calculate $c_i = C(S(\lambda+1,i))$ for $i = 1,\ldots,4$.

4. **Returning condition.** If $\sum_{i=1}^{4} c_i \geq c_0$ then $S(\lambda, \theta)$ should not be decomposed.[4] Reinstate the original $S(\lambda, \theta)$ from the stored copy. Add a QT node indicating that decomposition did not occur. Return from this level of recursion.

5. **Recursion.** Since execution passed through step 4, $S(\lambda, \theta)$ should be decomposed. Add a QT node indicating that decomposition occurred. If $\lambda \geq \lambda_{\max}$ return from this level of recursion. Otherwise, recursively call step 2 with new sub-bands $S(\lambda + 1, i)$ for $i = 1, \ldots, 4$.

The above algorithm details only the wavelet packet decomposition, which is part of the compression phase. During decompression, the inverse wavelet packet transform can be performed by doing a depth-first traversal of the QT, convolving the columns and then the rows of sub-bands with the inverse wavelet basis matrices.

An important step in the algorithm (and that which distinguishes wavelet packets from the standard wavelet transform) is step 3 in which the stopping criterion is calculated. This step represents the decision to further decompose a sub-band. We develop several measures which can be used to calculate decomposition stopping criteria.

3.4.2 Stopping Criteria

Recall from Section 2.2 that the purpose of the mapper is to produce a distribution of wavelet coefficients with minimal entropy. One inspiration for using wavelet packets is to allow a more flexible decomposition that will result in significantly more small coefficients, on average, than with regular pyramidal wavelet decomposition. The decomposition algorithm must know when *not* to decompose a sub-band and this is the responsibility of the stopping criterion.

We develop four related criteria which attempt to measure the effectiveness of packet decomposition: energy, entropy, kurtosis, and insignificance. All measures are based on the values of wavelet coefficients in a given sub-band. Note that in the following discussion "parent" will represent the sub-band being examined for possible decomposition and "children" will represent the four resulting sub-bands from such a decomposition.

Energy

Suppose that $c(j, k)$ returns the wavelet coefficient at location (j, k) in a given $M \times N$ sub-band. The energy of a sub-band $E(S(\lambda, \theta))$ is simply the sum of the absolute

[4] The comparison operator may actually be $\leq$ depending on the stopping criterion used. See Section 3.4.2 for more details.

values of the transform coefficients [10]:

$$E(S(\lambda, \theta)) = \sum_{j,k \text{ in } S(\lambda,\theta)} |c(j,k)|. \tag{3.12}$$

A small energy will correspond to a large number of small coefficients which suggests a highly-peaked coefficient distribution. Decomposition should halt if the sum of the energy in the children is greater than or equal to the parent's energy.

Entropy

Recall from equation (2.1) on page 8 that the entropy H of a source which has a fairly uniform spread of values will be high. Transforming the source into a highly-peaked distribution will imply minimizing the entropy. This stopping criterion is calculated on each sub-band by first building a coefficient histogram which counts the occurrence of each coefficient rounded to its nearest integer. This histogram is then normalized based on the total number of coefficients in the sub-band to yield a probability histogram. The entropy H can then be calculated. In order to be able to compare the parent's entropy with that of the children, H is first normalized by multiplying by the coefficient count.

$$H = MN \times \left(-\sum_{i} p(i) \log_2 p(i) \right). \tag{3.13}$$

Again, decomposition should halt if the sum of the entropy in the children is greater than or equal to the parent's entropy.

Kurtosis

Kurtosis is a statistical measure of a probability distribution's similarity to the normal distribution.[5] In our case, we want each sub-band to be as *dissimilar* to the normal curve as possible to maximize compression efficiency. Kurtosis K is defined as

$$K(S(\lambda, \theta)) = \left(\frac{1}{MN} \sum_{j,k \text{ in } S(\lambda,\theta)} \left(\frac{c(j,k) - \mu}{\sigma} \right)^4 \right) - 3, \tag{3.14}$$

where μ is the sample mean and σ is the standard deviation of $S(\lambda, \theta)$. Using kurtosis, decomposition should halt if the average kurtosis of the children is less than or equal to the parent's kurtosis.

[5] No assumption of normality need be met to calculate the kurtosis.

Insignificance

The final stopping criterion is based on the insensitivity of the HVS to small wavelet coefficients. Such coefficients are considered to be insignificant. This measure is a simple count of the number of wavelet coefficients below a certain magnitude. The insignificance I of a sub-band is then

$$I(S(\lambda,\theta)) = \sum_{j,k \text{ in } S(\lambda,\theta)} \begin{cases} 1, & \text{if } |c(j,k)| < c_I \\ 0, & \text{otherwise} \end{cases} \qquad (3.15)$$

where c_I is the minimal significant coefficient magnitude. The value c_I is set using experiments conducted by Watson *et al.* [58, 59] on the HVS sensitivity to errors in wavelet coefficients, which is described in Section 3.5.1 with quantization.

The less significant a sub-band is, the easier it will be to compress. Therefore, our goal is to maximize the insignificance of each sub-band. In this case, decomposition should halt if the insignificance of the children is less than or equal to the parent's insignificance.

Regardless of which stopping criterion is used, the decision to decompose a sub-band must be recorded so that the inverse wavelet packet transform will be performed in the correct order on the appropriate sub-bands. This information is stored in the QT data structure.

3.4.3 Quadtree Structure

The QT structure used in the decomposition algorithm explained in Section 3.4.1 deserves more examination. The QT can actually be easily stored as a bit string built dynamically from the recursive wavelet packet decomposition. A sub-band being decomposed could be represented by a "1" flag and a sub-band that does not need further processing could be indicated by a "0" flag. Note that the wavelet packet decomposition will build the QT in a depth-first traversal. Hence both the encoder and decoder know that the order of decomposition is depth-first and at any given level the four sibling nodes are recorded in the order of the sub-bands ($\theta = 1, \ldots, 4$).

As an example, the QT structure shown in Figure 3.6 would be represented as the bit string

1 1 1 0000 1 0000 00 1 001 0000 0 00.

The spaces are added for clarification to indicate a change in level λ. Recall that the maximum depth of the tree, which is specified as λ_{max}, is also known *a priori*.

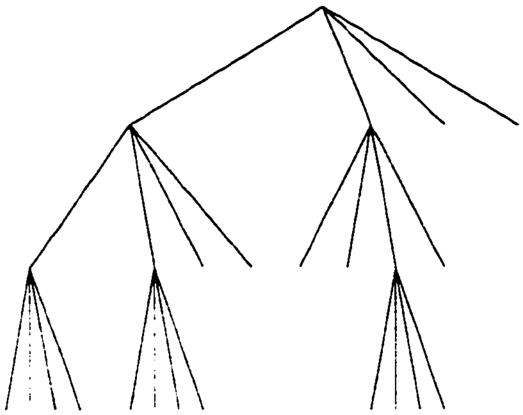


Figure 3.6: An example wavelet packet decomposition quadtree structure.

Therefore, there is no need to record the leaf nodes at the bottom level of the tree. So our example can be shortened to

$$1\ 1\ 1100\ 1\ 0010\ 00.$$

The number of sub-bands S_n in the decomposition can be simply calculated using the decomposition flags ("1") count,

$$S_n = \left(\sum_{\text{Bit string}} \begin{cases} 1, & \text{if bit flag is a "1"} \\ 0, & \text{otherwise} \end{cases} \right) \times 3 + 1. \tag{3.16}$$

The maximum length of the QT bit string will occur when all sub-bands are decomposed to the maximum depth λ_{max}. In this case we will have a string consisting of

$$\sum_{\lambda=0}^{\lambda_{\text{max}}-1} 4^{\lambda} \tag{3.17}$$

ones, since each increase in λ decomposes each sub-band into four new bands. For example, $\lambda_{\text{max}} = 4$ will produce a string of 85 ones. This bit string requires a QT structure "overhead" of approximately 4.1×10^{-5} bpp for a 512 × 512 pixel image. The extra bits needed to describe the QT structure are negligible, but there is computational overhead involved in using a packet decomposition. Additionally, as the number of sub-bands grows, the number of quantizer parameters which need to be stored also increases. We shall see in Chapters 4 and 5 if the increased compression capability of more sub-bands will outweigh the extra bits required for describing each quantizer.

3.4.4 Complexity Issues

The wavelet packet decomposition incurs additional computational load over the typical pyramidal transform. The stopping criterion must be calculated for each sub-band (except for when λ_{max} is reached) and the convolution operations will be performed more often. However, this extra computation is not nearly as much as would be required for a typical bottom-up wavelet packet approach, in which the "optimal" decomposition is found [48, 65].

In an optimizing wavelet packet decomposition, the convolution operations must be performed until λ_{max} is reached for every orientation θ. Next, an optimality parameter will have to be re-calculated for every possible change in the "pruned" decomposition tree. This step will also entail possible inverse wavelet transforms to reconstruct certain sub-bands. unless extra memory is used to store all possible decompositions.

Once the wavelet packet transform has been completed, the next stage in the WP/HVS framework is to quantize the wavelet coefficients. This stage involves loss of information, but also the greatest increase in compression.

3.5 Quantization

The framework includes the ability to quantize the wavelet coefficients using either a uniform or non-uniform scalar quantizer. Vector quantization (VQ) is not supported for two main reasons:

1. The wavelet transform is designed to remove as much of the second-order correlation as possible. A VQ scheme may not produce improved results over a scalar technique since it examines blocks of wavelet coefficients. The FBI rejected VQ from their wavelet-based codec since it did not improve performance over scalar quantization [5].

2. Codebook-based VQ is computationally-intensive and requires extra memory for storing the codewords. Recall from Chapter 1 that simplicity is one of our final goals. We need to produce a system that would not require copious hardware support, since our proposed platform will be mobile computers. Lattice

vector quantization does not require codebook design, and this technique used in conjunction with wavelet compression has already been examined by other research members involved in the Motorola project [33]. It was found that LVQ could provide a small increase in PSNR. However. visible quality decreased, particularly in "smooth" image areas.

In order to take advantage of the different coefficient distributions and the varying sensitivity of the HVS, a quantizer with unique parameters is used on every sub-band. All sub-bands excepting the LL band $S(\lambda_{\max}, 1)$ use quantizers centered on zero, since this is the most common coefficient. Both uniform and non-uniform scalar quantizers used in the WP/HVS framework incorporate HVS characteristics. However, the non-uniform quantizer has the added capability of directly modeling visual masking (see Section 2.7.3). There are certain characteristics which both quantizers share.

3.5.1 A Quantizer Stop-Band Based on the HVS

As mentioned in Section 3.4.2 in relation to the wavelet packet decomposition stopping criteria. very small wavelet coefficients represent image regions of low activity, and are insignificant to the HVS. Watson *et al.* conducted experiments to determine an error threshold for wavelet coefficients below which the distortion is invisible to a human observer [58, 59]. They propose that the visible distortion thresholds could be used in designing a wavelet codec such that no quantization errors are visible. In this case. they dub the reconstructed image to be "perceptually lossless" indicating that the results are *numerically* different but *visually* the same as the original image.

The experiments were conducted using images of uniform intensity. Uniformly sampled errors in the interval $[-1, 1]$ were added to the wavelet coefficients in an individual sub-band, with all other coefficients being held at zero. The reverse wavelet transform was then applied to yield an image with quantization noise stimulus. Images with and without noise stimulus were shown to human observers for a specific length of time. The observers were instructed to select which image contained quantization noise. The amplitude of the noise signal could be scaled, so that a visibility threshold could be found.

After recording the sensitivity of observers to certain amplitudes of quantization noise, Watson *et al.* use their experimental data to construct a mathematical model of the visibility thresholds based on level λ and orientation θ of the wavelet sub-bands. The model can be used to calculate a set of uniform quantization factors. These quantization factors indicate how wide the uniform bins can be without producing noticeable distortion. For our uniform quantizer, the insignificance threshold c_I can be found by halving the Watson quantization factors, since this would be the maximum error introduced by a uniform quantizer. Table 3.1 lists the insignificance thresholds for the wavelet sub-bands.

	Orientation θ			
Level λ	**1**	**2**	**3**	**4**
1	7.0	11.5	29.4	11.5
2	5.6	7.3	12.2	7.3
3	5.7	6.4	9.8	6.4
4	7.3	7.1	8.9	7.1

Table 3.1: Values of the insignificance threshold c_I for varying level λ and orientation θ in the wavelet packet decomposition.

Notice that the HVS is most sensitive to errors in orientation $\theta = 1$, less sensitive in $\theta = 2$ or 4. and least sensitive in orientation $\theta = 3$ which holds the diagonally-oriented signals. These results concur with what we would expect from such an experiment, given our discussion of the contrast sensitivity function (CSF) in Section 2.7.2.

We can use c_I in setting all coefficients $c(j, k)$ in a sub-band to zero if their magnitude is less than c_I. Therefore, $\pm c_I$ sets the limits of the quantizer "stop-band". We can also use c_I to set the size of the remaining quantizer bins (i.e., the width of each will be $2c_I$) since this distortion should not be visible.

Recall, however, that the quantization factors were found by Watson *et al.* using quantization noise in otherwise *uniform* images in experiments, which ignores the effects of visual masking. The bin sizes for coefficients reflecting active signal content may be set larger without introducing perceptual distortion.

3.5.2 Modeling Signal Content Sensitivity

We can attempt to take advantage of the HVS characteristic that errors are harder to see in areas of greater signal content by using a non-uniform quantizer. In this case, as the wavelet coefficients get larger, they can be quantized more coarsely, since the greater distortion will be masked by the more intense signal.[6]

[6] The masking effect will not be modeled in the highest decomposition level $S(\lambda_{\max}, 1, \ldots, 4)$ since these sub-bands are low-resolution approximations of the image, showing large-scale changes in intensity.

Significantly, the main design goal of a non-uniform quantizer is to minimize the mean-square error (MSE). To do so with wavelet coefficients requires larger quantizer bins in areas of larger coefficient magnitude, since these values have lower probability than smaller values. This characteristic is perfectly-suited to take advantage of masking, since larger coefficients imply a busier signal.

Designing the Quantizer

As mentioned in Section 2.5.2, it is difficult to design an optimal non-uniform quantizer under most circumstances. Our situation is unique, however, in that we know *a priori* what the output levels and bin limits should be around the zero value by using the stop-band discussed in Section 3.5.1. Non-uniform quantizers are usually implemented using companding. otherwise the first output level q_1 must be found by a trial-and-error process. In our case, the entire companding process is unnecessary since the optimal (in an HVS-relevant sense) value for the first output level q_1 is known. The minimum bin size is also known and the bins should increase in size with increasing coefficient magnitude to take advantage of the masking effect. The non-uniform quantizer is therefore guaranteed to result in fewer output levels than the uniform quantizer.

Examine Figure 3.7. The non-uniform quantizer can be designed by symmetry. That is, the output levels and bin limits can be found on one half of the distribution and mirrored on the other half. The values z_1 and z_2 are known, since they are defined to be $\pm c_I$. The first output value q_1 is zero. According to equation (2.11), q_2 can also be immediately found since

$$q_k = 2z_k - q_{k-1}, \tag{3.18}$$

$$q_2 = 2z_2 - q_1. \tag{3.19}$$

$$q_2 = 2z_2. \tag{3.20}$$

The remainder of the design process requires the determination of $q_3, \ldots, q_K$ and $z_3, \ldots, z_{K+1}$.

Calculating the remaining bin limits can be done indirectly using a discrete (i.e., non-continuous) version of equation (2.12),

$$q_k = \sum_{i=z_k}^{z_{k+1}} \frac{ip(i)}{p(i)}. \tag{3.21}$$

In this case, we first need to build the probability histogram $p(i)$ of the wavelet coefficients in this sub-band. Every variable in equation (3.21) is known except for

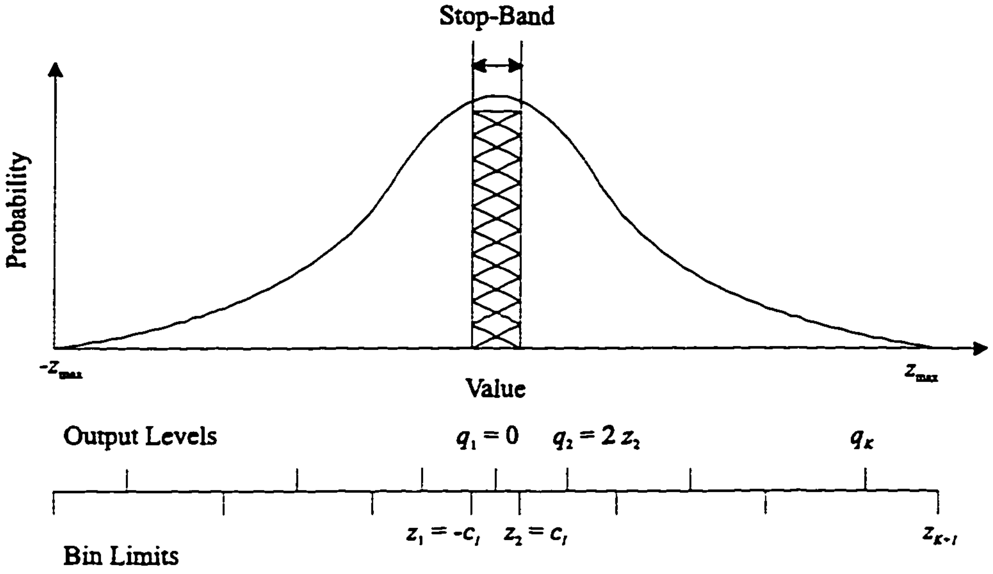


Figure 3.7: Designing the non-uniform quantizer.

z_{k+1}. Therefore, the summation should be performed until q_k is reached, at which point z_{k+1} has been calculated. Finding the next output level is then a simple matter of applying equation (3.18) with the newly found z_{k+1} in place of z_k. Once z_{k+1} is found to be equal to or beyond the maximum coefficient $z_{\max}$, we set $z_{k+1} \leftarrow z_{\max}$ and stop the procedure. Note that the final q_{K} from this process is an unreliable value, due to the truncation of z_{K} to $z_{\max}$ and is not used for the modeling step explained below.[7]

All z_k and q_k are now known. Sending all of these parameters to the decoder for each sub-band would amount in a significant amount of overhead. It is a better idea to model q_k with a simple polynomial, reducing the required overhead to the polynomial coefficients.

[7] If the design procedure produces three or fewer valid output levels, a default uniform quantizer is used instead.

Modeling the Quantizer Parameters

We have chosen to model q_k with a parabola,[8]

$$q_k = \beta_0 + \beta_1 k + \beta_2 k^2. \tag{3.22}$$

Only two parameters are then required to model the output levels of each quantizer. Is a quadratic a polynomial of too low an order to precisely represent the changing bin size? Recall that all q_k are already approximates, due to the discrete nature of the algorithm finding q_k. Rounding is involved in building the probability histogram and for the summation finding each z_k. In addition, the exact shape of the masking function is not known, so as long as we have increasing bin size with increasing signal content, we can attempt to model masking [19].

The implementation of the parabolic model uses curve-fitting in a least-squares sense. This process can be accomplished through simple matrix operations. First, we set up the matrices $\mathbf{Q}$ and $\mathbf{K}$,

$$\mathbf{Q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ \vdots \\ q_K \end{bmatrix}, \quad \mathbf{K} = \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 2 & 4 \\ \vdots & \vdots \\ q_K & q_K^2 \end{bmatrix}. \tag{3.23}$$

Then we solve the equation

$$\hat{\beta} = (\mathbf{K}^t\mathbf{K})^{-1}\mathbf{K}^t\mathbf{Q} \tag{3.24}$$

in which $\hat{\beta}$ is the matrix holding the model coefficients,

$$\hat{\beta} = \begin{bmatrix} \widehat{\beta_1} \\ \widehat{\beta_2} \end{bmatrix}. \tag{3.25}$$

The parameters $\widehat{\beta_1}$ and $\widehat{\beta_2}$ will have to be sent to the decoder so that it can properly assign the quantized coefficients.

An example set of quantizer output levels and the computed model function is shown in Figure 3.8. There are 7 output levels. Notice how the 7th value is far off the trend, as was discussed above. This value was ignored in the modeling process. We therefore have a *conservative* model in the sense that we introduce less error in the largest coefficients than allowable, according to the data points. It is obvious that the model does not precisely fit all data points, but the trade-off of not having to send all the output levels is in our favor.

[8] Note that we will always assume $\beta_0 = 0$ in equation (3.22) because the y-intercept (i.e., q_1) is always 0.

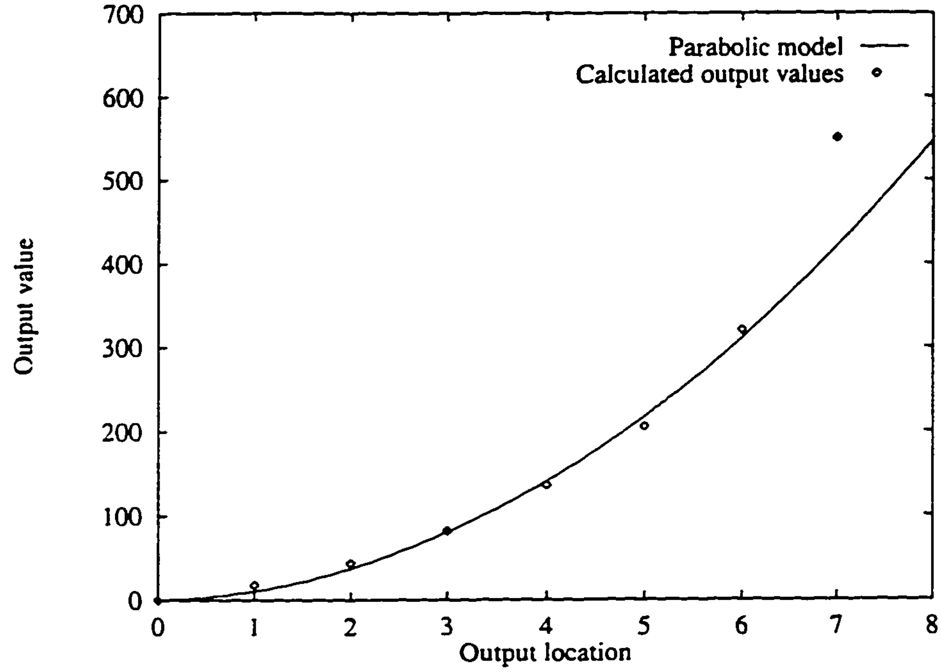


Figure 3.8: Example non-uniform quantizer output levels and their parabolic model.

The modeling process may require the number of quantizer bins K' to be increased in order to cover the full range of wavelet coefficients. (For example, note that the model in Figure 3.8 would predict $q_{K'}$ to be about 420 when it was calculated to be 550.) The new K' can be easily found by setting the parabolic model equal to the previously found $q_{K'}$,

$$\widehat{\beta_1} K' + \widehat{\beta_2} K'^2 - q_{K'} = 0, \tag{3.26}$$

and then solving for K' using the quadratic formula,

$$K' = \frac{-\widehat{\beta_1} + \sqrt{\widehat{\beta_1}^2 + 4\widehat{\beta_2} q_{K'}}}{2\widehat{\beta_2}}. \tag{3.27}$$

For the example in Figure 3.8, the new K' is found to be 8, yielding $q_{K'} = 547$, which is more accurate than our previous calculation. Recall that we are dealing only with the positive coefficient range, so the number of bins is actually $2K' - 1$.

Quantization and Dequantization

The quantization process using the non-uniform quantizer is slightly more complicated than with the uniform quantizer. First, we have to calculate each output value q_k given the quadratic model. Then, each wavelet coefficient is assigned to its closest q_k (in an absolute difference sense), also recording whether the value is positive or negative. Dequantization, or finding the quantized values of the wavelet coefficients during decoding, is a simple process of calculating equation (3.22) with each decoded q_k and the quantizer's parabolic model parameters.

3.5.3 Acceptable Distortion Factor

The application of either scalar quantizer of the WP/HVS framework will result in a "perceptually lossless" image at a minimum possible bit-rate. There may be occasions upon which noticeable distortion in the reconstructed image will be tolerated by the user in order to further reduce the bit-rate. To facilitate greater compression, an acceptable distortion factor (ADF) is introduced.

The ADF is implemented as a quantity which modifies the quantizer stop-band size and, therefore, the remaining bin sizes. This factor is multiplied by the insignificance threshold c_I of each sub-band. That is,

$$c_I(\lambda, \theta) \leftarrow \text{ADF} \times c_I(\lambda, \theta). \tag{3.28}$$

Applying the ADF in this manner ensures that perceptually less-significant sub-bands will increase in distortion faster than more important bands.

The default is ADF = 1.0 and if the user chooses to increase it, the result will be larger distortion and greater compression. Note that this system is arguably more user-friendly than setting the exact bit-rate, which many codecs enforce [53, 51, 64], and is superior to some schemes which do not allow flexible compression rates at all. The ADF is analogous to the quality factor in the JPEG standard [57], but is better than JPEG in that it mirrors HVS characteristics. Incorporating the ADF also eases the process of maximally compressing an image without noticeable degradation, since the ADF would not have to be modified by the user in this case.[9]

[9] "Perceptually lossless" coding could possibly be the most practical codec application. For example, most World Wide Web applications require high-quality images free of noticeable distortion.

3.5.4 Complexity Issues

The uniform quantizer presents the minimal computational complexity requirements. The bin limits (and therefore the output values) are set by a simple table look-up for the insignificance values. The quantization process must divide each wavelet coefficient by the number of bins K. Using the values from Watson *et al.*, along with the ADF, allows efficient quantization, especially compared with rate-distortion optimizing frameworks. The space overhead for each uniform quantizer consists of sending the maximum output level q_K and the number of bins K. Since the coefficient distribution for the $S(\lambda_{\max}, 1)$ sub-band is not zero-centered, the minimum output level q_1 must also be sent.

Applying the non-uniform quantizer increases the complexity, in both time and space overhead. Each q_k and z_k must be calculated, which involves creating the probability histogram and performing the summation of equation (2.12). The parameter modeling step requires several matrix operations and then the re-determination of q_K. Actually quantizing the coefficients will also take more time since each q_k needs to be calculated from the model. The parameters to be sent are q_K, $\widehat{\beta_1}$, and $\widehat{\beta_2}$. Note that K does not need to be transmitted because the dequantizer does not use it.

The complexity of the non-uniform quantizer is greater than that of the uniform quantizer, but is far smaller than most vector quantization schemes or rate-distortion optimizing scalar quantizers.

3.6 Quality Assessment Tools

Once the image has been compressed and reconstructed using a configuration of the WP/HVS framework, we need a set of techniques to assess the quality of the results. The WP/HVS framework supports three different methods for examining and comparing image quality, including peak signal-to-noise ratio (PSNR), frequency weighted PSNR, and distortion maps.

3.6.1 Peak Signal-to-Noise Ratio

Since PSNR is still the *de facto* standard for quality assessment, it would be irresponsible to not use it to report results.[10] Applying PSNR also gives one way to directly compare with most other results in the literature, since it is such a popular measure.

[10] See equation (2.15) in Section 2.6 for the definition of PSNR.

3.6.2 Frequency-Weighted Peak Signal-to-Noise Ratio

As discussed in Section 2.6, most researchers agree that PSNR is not a particularly accurate quality measure as it is normally defined. A simple improvement involves weighting spatial frequencies to concur with the HVS sensitivity, yielding a measure which should correlate more to an observer's experience.

Kim *et al.* have used a weighted mean-square error measure with HVS-based wavelet compression [31]. A picture quality scale (PQS) incorporates two factors based on weighted frequency errors [42].[11]

One of the factors is the CCIR 567-1 standard for quality analysis used by the television industry. To implement this factor, the first step is to find the error image $e(x,y)$ from the original and reconstructed images,

$$e(x,y) = f(x,y) - \hat{f}(x,y). \tag{3.29}$$

Then the error image is weighted according to its spatial frequencies by convolving it with a filter $w_1(x,y)$ in the Fourier domain,

$$e_{w_1}(x,y) = e(x,y) * w_1(x,y). \tag{3.30}$$

The CCIR standard defines the weighting filter to be

$$w_1 = \frac{1}{1+\left(\frac{\sqrt{u^2+v^2}}{f_c}\right)^2}, \tag{3.31}$$

where u and v are the horizontal and vertical spatial frequencies, respectively. The CCIR standard is implemented as a low-pass filter with cutoff frequency $f_c = 5.56$ cycles/degree.

PQS uses $e_{w_1}(x,y)$ to form a normalized MSE for one of the error factors. In order to form a more useful error measure when used independently, we can define a weighted PSNR (WPSNR) by using the weighted error image $e_{w_1}(x,y)$,

$$\begin{aligned} \text{WMSE}_1 &= \frac{1}{MN}\sum_{j=1}^{M}\sum_{k=1}^{N} e_{w_1}(j,k)^2, & (3.32) \\ \text{WPSNR}_1 &= 10\log_{10}\frac{255^2}{\text{WMSE}_1}. & (3.33) \end{aligned}$$

A second PQS factor also examines random errors, but the convolution is performed with a filter w_2 which takes into account the anisotropy of the CSF and is

[11] The other PQS factors deal mainly with end-of-block errors and errors on edges, which are not particularly important when judging wavelet codecs, but may be more important with spatial domain techniques or Fourier-based transforms due the their non-compact support.

modeled with $\mathrm{CSF}(u,v)$ of equation (3.11). Distortion which is below a perceptual threshold T is ignored.[12] In addition, the original and reconstructed images are gamma corrected. We can develop another WPSNR using this technique by the following,

$$\begin{aligned} e(x,y) &= \gamma(f(x,y)) - \gamma(\hat{f}(x,y)), && (3.34) \\ e_{w_2}(x,y) &= e(x,y) * w_2(x,y), && (3.35) \\ \mathrm{WMSE}_2 &= \frac{1}{MN}\sum_{j=1}^{M}\sum_{k=1}^{N}\begin{cases} e_{w_2}(j,k)^2, & \text{if } e_{w_2}(j,k) \geq T \\ 0, & \text{otherwise} \end{cases}. && (3.36) \end{aligned}$$

Then WPSNR_2 is defined similarly to equation (3.33).

3.6.3 Distortion Maps

The three quality measures (PSNR, WPSNR_1, and WPSNR_2) can give a quantitative judgment of reconstructed image quality. However, these measures do not indicate the *location* of any image distortion. Knowing the location of errors is important, since the HVS is sensitive to image signal content. The weighted PSNR measures weight errors based on their spatial frequencies, but the observer may weight errors based also on their location in the image.

We can use the three error measures to produce distortion maps. The maps are simply the squared error images resulting from the different measures. Pixels with larger errors will be displayed with greater intensity.[13] Using PSNR, the distortion map m will indicate each pixel's error by,

$$m_{\mathrm{PSNR}}(x,y) = \left(f(x,y) - \hat{f}(x,y)\right)^2. \qquad (3.37)$$

Similarly, the weighted distortion maps will be composed of the pixels,

$$m_{\mathrm{WPSNR}}(x,y) = e_w(x,y)^2, \qquad (3.38)$$

where e_w is a pixel's weighted error calculated from either equation (3.30) or (3.35).

[12] The developers of PQS set $T = 1.0$.

[13] We will display each distortion map by its *inverse*, since printing larger errors darker on paper yields more clarity.

3.7 Summary

Within this chapter, we have examined many of the components of the proposed WP/HVS framework, including:

- modeling the luminance and spatial frequency sensitivities of the HVS through pre-processing;
- the wavelet packet decomposition and proposed stopping criteria;
- the structure of both uniform and non-uniform scalar quantization, including our proposed method of modeling the signal content sensitivity of the HVS:
- and the quality assessment tools.

There are many decisions that need to be made about the WP/HVS *framework* in order to choose one configuration to form the WP/HVS *codec*. In the subsequent chapter, we will independently test each major framework component in the process of designing the new WP/HVS codec.

Chapter 4

Component Testing and Design of the WP/HVS Codec

4.1 Introduction

We will now attempt to design a high-performance, practical, general-purpose codec configuration based on the WP/HVS framework presented in Chapter 3. We will approach this task using a "greedy" design method, in which we analyze each significant component of the framework to find its most appropriate use. The steps in the design process will analyze:

- the experimental environment, including test images;
- human visual system (HVS) luminance sensitivity modeling;
- HVS spatial frequency sensitivity modeling;
- wavelet packet decomposition schemes;
- and quantization using uniform and non-uniform quantizers, the latter of which attempts to model the HVS masking effect.

The "best" components will then be assembled to form the new WP/HVS codec.[1] Certainly the resulting codec cannot be described as optimal. However, this design method will allow the staggered production of a codec without juggling all changeable parameters at once.

The three *objective* quality measures (PSNR, WPSNR$_1$, and WPSNR$_2$) presented in Section 3.6 will be applied to assess the results of each component during the

[1] The performance of the WP/HVS will be tested against other standard methods in Chapter 5.

design phase. Though the human observer is the most relevant quality assessor, it would be difficult to *subjectively* assess the results of all possible component parameter combinations. We would therefore like to choose the configuration of each component based on objective measures so the final codec has a solid theoretical foundation. The experimental results will also be easy to present and should be repeatable by other parties.

4.2 Implementation and Testing Environment

All experiments were run using C++ and C programs developed on the UNIX operating system. The experiments were executed on a 75MHz Pentium-based PC running Linux. Please see Appendix A for a brief description of the software implementation details. The wavelet packet transform employs filters used by Antonini *et al.* which have been shown to have high performance in image compression [3].[2]

4.2.1 Test Images

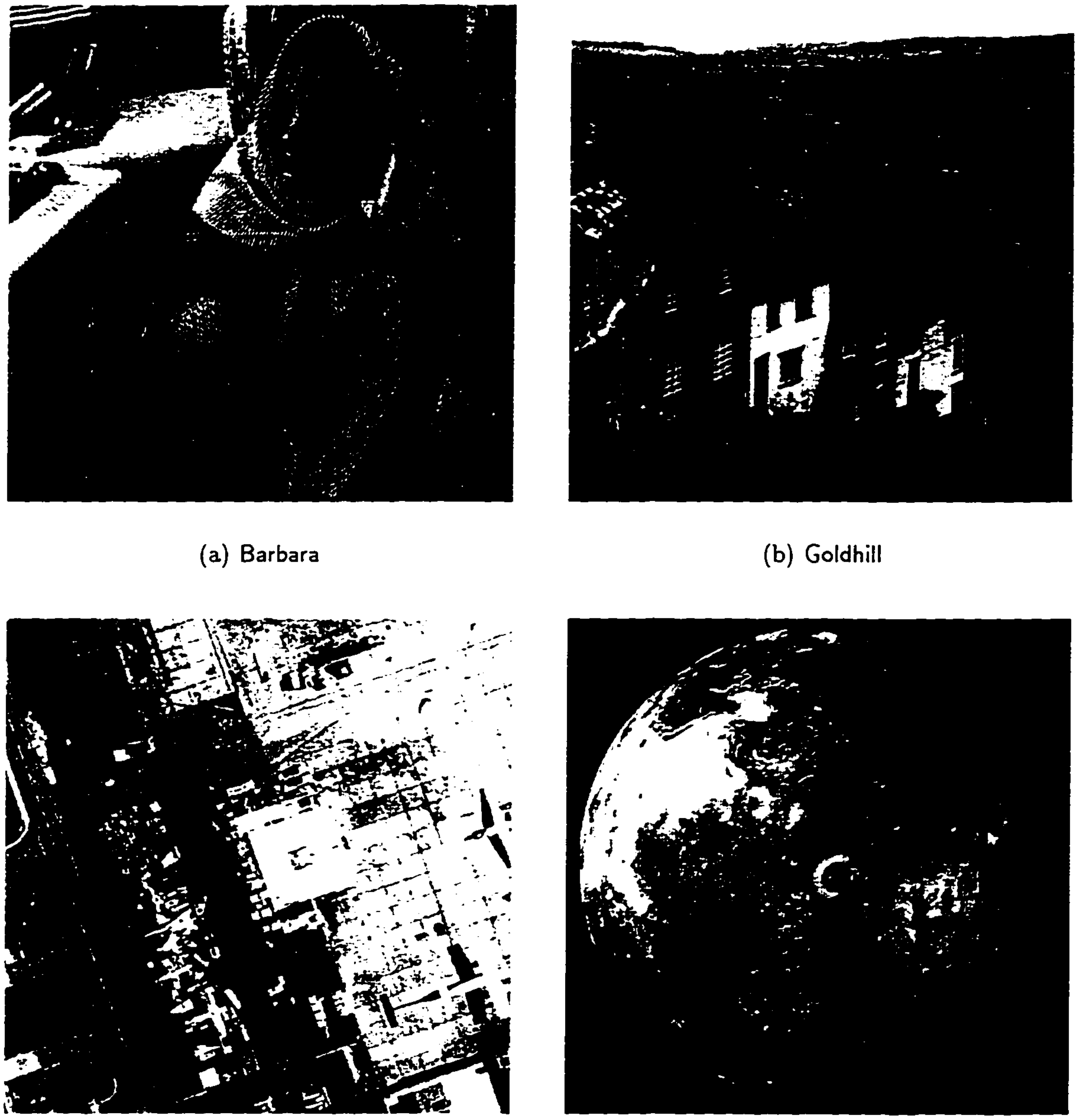
The image set used in the WP/HVS component experiments consists of 54 grayscale images grouped into five categories: Natural (18), Random Texture (8), Structured Texture (8), Fingerprint (10), and Ultrasound (10). All but the Ultrasound images have a 512 × 512 resolution. This categorization is for the convenience of discussing experimental results.

The 18 Natural images consist of a set of frequently used images in the image compression literature. The set includes scenes of people, food, animals, landscapes, and some aerial photographs. Four examples of this type of image are shown in Figure 4.1.

The texture images are 512 × 512 cropped versions of images scanned from the Brodatz texture album [7]. Examples of Random Texture and Structured Texture images are shown in Figure 4.2.

The Fingerprint category is a set of ten, rectangular, publicly available FBI fingerprint images. The original fingerprint images have been cropped to a 512 × 512 resolution in order to exert greater control and consistency over the experiments. We also have a set of ten Ultrasound images taken from live pigs. All Ultrasound images have a resolution of 234 rows × 260 columns. Example Fingerprint and Ultrasound images are shown in Figure 4.3.

[2] Exploring the design and application of different wavelet filters is beyond the scope of this thesis.



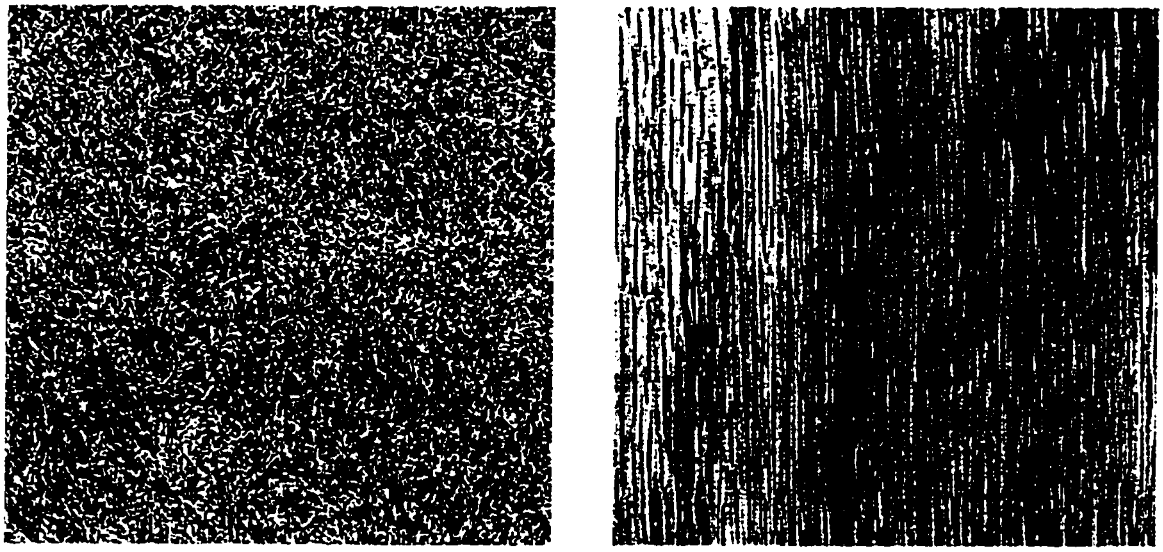
(a) Barbara

(b) Goldhill

(c) Airfield

(d) Io

Figure 4.1: Example Natural test images.



(a) Texture08

(b) Texture15

Figure 4.2: Example (a) Random Texture and (b) Structured Texture test images.

(a) Finger03

(b) Ultrasound05

Figure 4.3: Example (a) Fingerprint and (b) Ultrasound test images.

It should be stressed that our experimental image library contains a great variety and number of images, more than normally used for reporting image codec testing results [13]. Most of the image compression literature reports on results from very few images of one type (e.g., natural). Applying and reporting on a more thorough examination is more illustrative of strengths and weaknesses of a codec.

4.2.2 A Technique for Results Analysis

In subsequent sections, we will be analyzing experimental results of several configurations of the WP/HVS framework by comparing the performance based on the quantitative quality measures. This analysis will involve examining the average performance over categories, as opposed to individual images. It would be difficult to judge the relative efficacy of the various configurations based only on quality values, since we will be analyzing images compressed at *similar* but not *identical* bit-rates. One could not apply a standard statistical test to this data to check for significant performance differences, since both bit-rate and quality vary. We need to manipulate the data to give *one* dependent variable. We can attempt to "normalize" the bit-rate and quality by finding the average quality Q to bit-rate R ratios for all N images in a given category. That is,

$$\overline{Q/R} = \frac{1}{N}\sum_{i=1}^{N}\frac{Q_i}{R_i}. \tag{4.1}$$

This method will allow us to compare results of different WP/HVS framework configurations on a level playing field, but we do not claim it to be optimal in distinguishing the performance differences.

The Q/R ratio was inspired by the fact that a scheme with *minimal* bit-rate at *maximal* quality yields the best performance. To illustrate the Q/R ratio, Figure 4.4 shows a rate-distortion graph used to compare four fictitious results. It is clear that Technique 2 ($Q/R = 27.675/0.205 = 135$) is better than Technique 3 ($Q/R = 24.15/0.21 = 115$), since Technique 2 yields a higher PSNR *and* a lower bit-rate. Although it is not immediately obvious, Techniques 1 and 4 have similar performance ($Q/R = 19.5/0.15 = 28.6/0.22 = 130$).

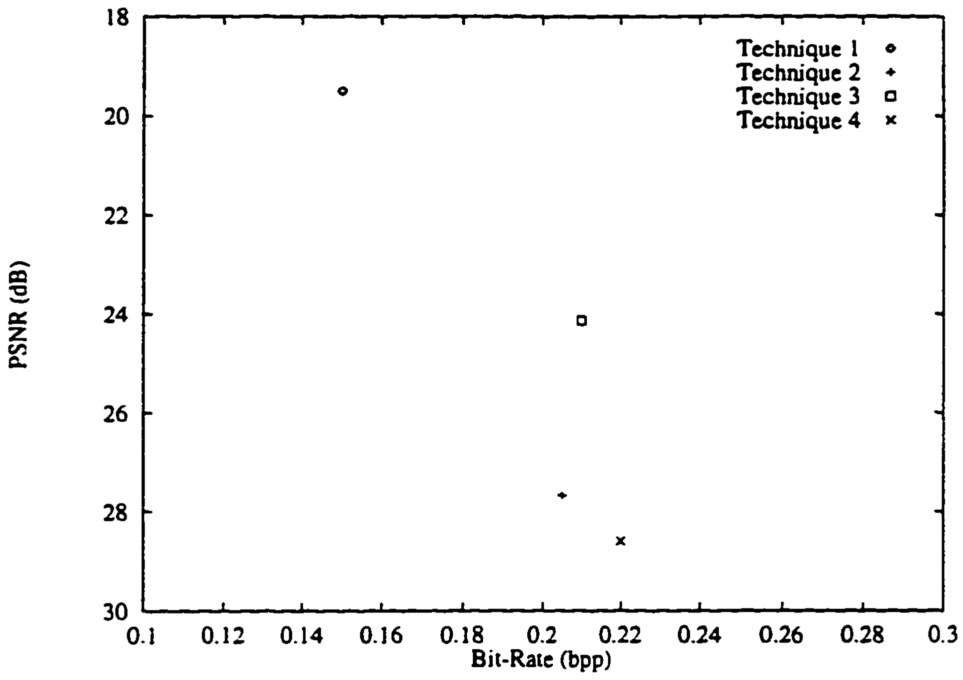


Figure 4.4: Example results from four fictitious compression techniques, used for demonstrating the Q/R ratio.

4.3 Gamma Correction Component Testing

A simple experiment was performed to assess the effects of applying gamma correction in the image compression process. Recall from Section 3.2 that gamma correction can be used to model the non-linearity of the HVS luminance sensitivity. Testing this component involved using the WP/HVS framework without CSF filtering and with uniform quantization.

It was found that incorporating gamma correction decreases the bit-rate at a given acceptable distortion factor (ADF). To achieve similar bit-rates when compressing with or without gamma correction, we used ADF = 1.0 with gamma correction and ADF = 1.7 without. The bit-rate is decreased by gamma correction since this process modifies the distribution of the original image pixel intensities. For example, Figure 4.5 shows the histograms of the Airfield image before and after gamma correction using $\gamma = 2.2$. Notice that gamma correction transforms the distribution, making it more peaked with a smaller dynamic range. On Airfield, gamma correcting decreases the entropy from 7.12 to 6.77 bpp, indicating that greater compression will likely be

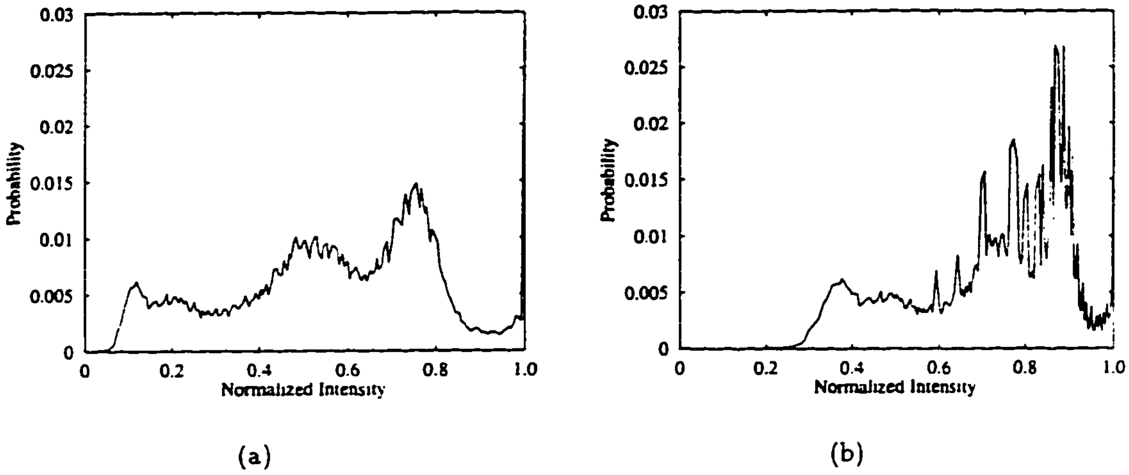


Figure 4.5: Normalized pixel intensity histograms from (a) original and (b) gamma-corrected ($\gamma = 2.2$) Airfield images.

achieved on the gamma-corrected image.

All test images were compressed with and without gamma correction and a summary of the results is shown in Table 4.1.

Image	With Gamma Correction			
Category	Bit-Rate	PSNR/R	WPSNR$_1$/R	WPSNR$_2$/R
Natural	0.636	50.20	54.24	65.67
Random Texture	1.647	13.20	14.36	19.82
Structured Texture	1.018	27.48	32.04	39.41
Fingerprint	0.617	39.78	42.26	53.51
Ultrasound	0.338	98.04	109.80	109.93
	Without Gamma Correction			
	Bit-Rate	PSNR/R	WPSNR$_1$/R	WPSNR$_2$/R
Natural	0.531	**67.87**	**87.46**	**86.34**
Random Texture	1.653	**16.13**	**24.44**	**26.35**
Structured Texture	0.938	**33.18**	**46.59**	**48.46**
Fingerprint	0.646	**52.44**	**65.87**	**68.15**
Ultrasound	0.307	**107.62**	**121.48**	**119.25**

Table 4.1: Experimental results from compressing images with and without gamma correction pre-processing.

The table shows the Q/R ratios for PSNR, WPSNR$_1$, and WPSNR$_2$, averaged over image category. The superior results are shown in bold. When compressed to similar bit-rates, Table 4.1 shows that the quantitative quality measures indicate that the gamma-corrected results are worse than without gamma correction. From visual inspection, the quality of the reconstructed image with or without gamma correction appears to be insignificantly different.

This experiment supports Clarke's statement that the effects of incorporating the HVS non-linearity into the codec

> *tend to be over-ridden by the major non-linearity in the processing chain, and this is the conventional display monitor, where the light output / electrical signal input relationship has an exponent of something like 2.5.* [13].

In other words, most computers already incorporate gamma correction in displaying graphical information, either in the video buffer or the monitor itself. The video buffer can be inverse-gamma transformed, which will appear correct to the HVS using a monitor which is *not* gamma-corrected. Another strategy is using an electron gun in the monitor which has an inverse-gamma response. The forward-gamma response of the HVS will compensate. In either case, the intensities are sampled such that each pixel's gray level is approximately proportional to perceived brightness.

In conclusion, incorporating gamma correction into compression is a waste of computation, unless the display hardware is not built to compensate for the non-linearity of the HVS. It would be challenging to find modern hardware lacking this compensation.

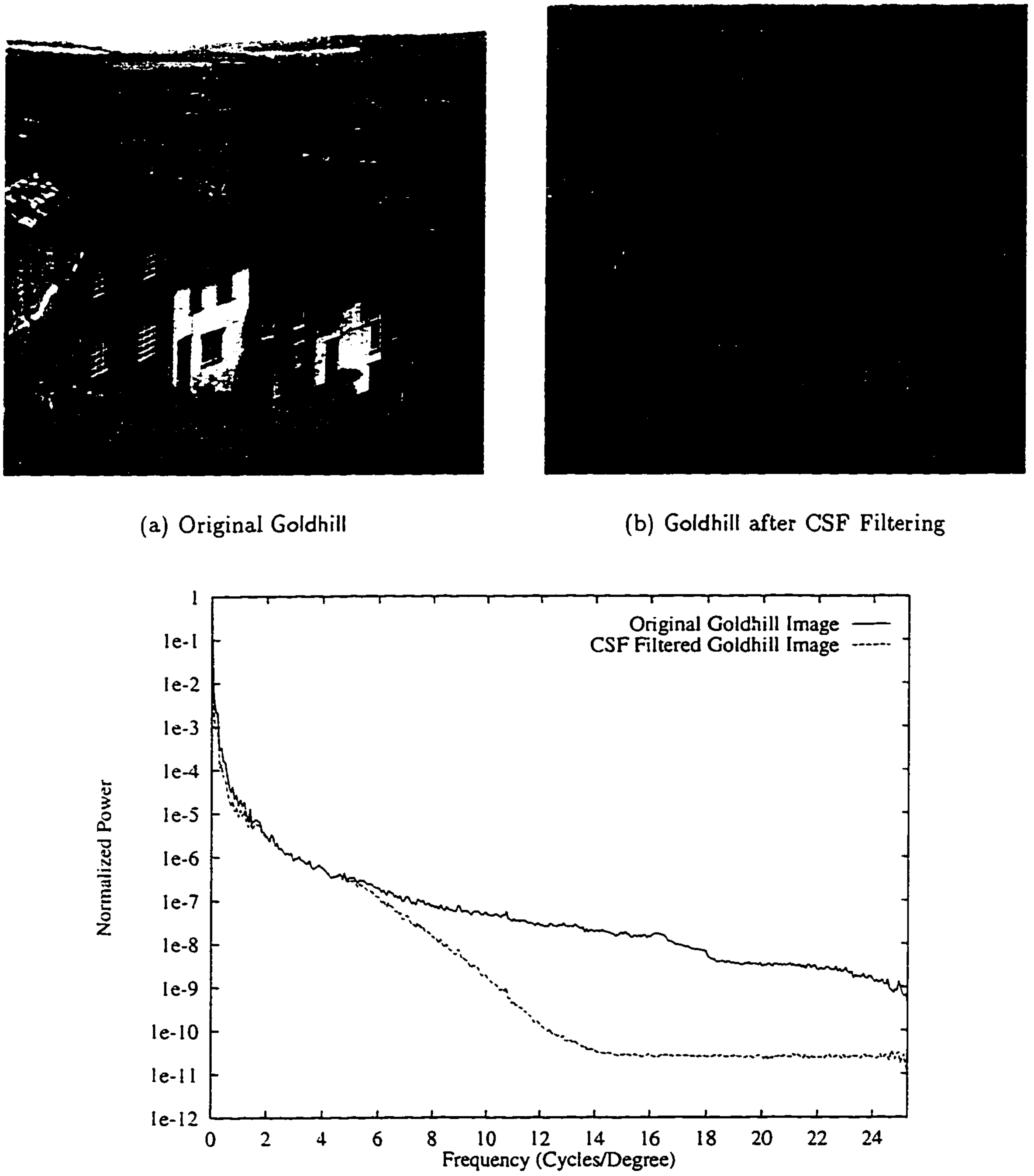
4.4 CSF Filtering Component Testing

The WP/HVS framework supports another optional pre-processing step, which attempts to filter out data of spatial frequencies to which the HVS is insensitive. This is a lossy procedure, but the purpose is to yield an image which is perceptually lossless (see Section 3.5.1). However, any quantitative quality measure will indicate a difference from the original image.

The anisotropic model of the contrast sensitivity function (CSF) filter reported in Section 3.3 can be directly applied. We can set the near and far viewing distance range to be a typical $d_n = 4h$ and $d_f = 6h$, where h is the image height. This procedure applied to the Goldhill image yields the result shown in Figure 4.6 (b).

As suggested from the figure, using this band-pass filter reduces the image mean significantly. Note that while the CSF filtered image is recognizable as Goldhill, the PSNR is a low 14.81. For comparison, a uniform image of intensity 113 would yield the same PSNR![3] The drop in image mean is caused by filtering out some of the low-pass energy, which is highly concentrated in most images. The 1D averaged power spectra of both images are shown in Figure 4.6 (c).

[3] Certainly this result supports the critics of PSNR.



(a) Original Goldhill

(b) Goldhill after CSF Filtering

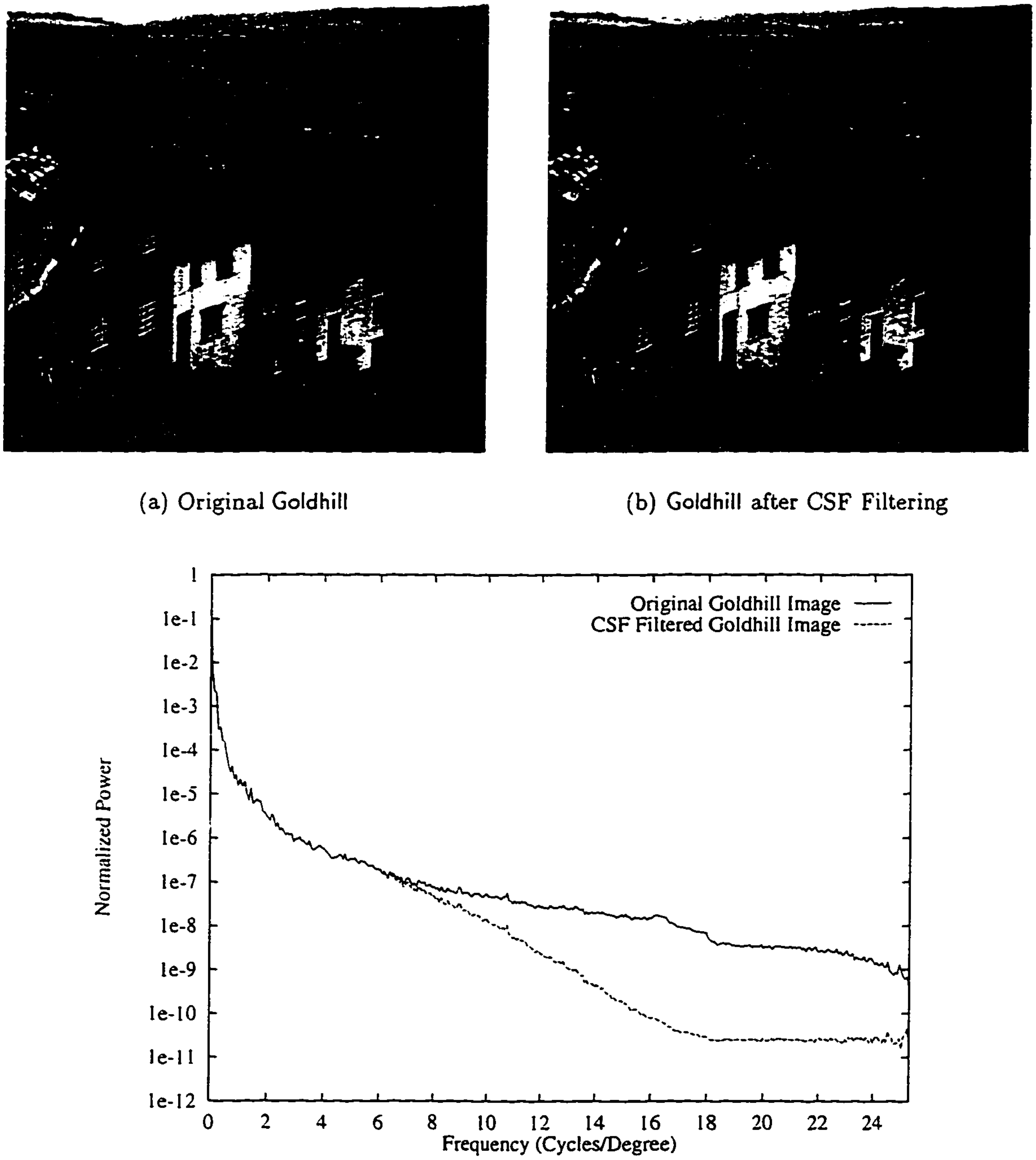
(c) Power Spectra of (a) and (b)

Figure 4.6: Applying the CSF filtering operation to Goldhill with $d_n = 4h$ and $d_f = 6h$.

Experiments have found that the HVS frequency sensitivity falls off rapidly in the low-pass spectrum (i.e., less than 4 cycles/degree). These experiments were carried out to find a threshold of sensitivity. However, applying the CSF directly in image compression can cause problems, since we need to compare the results with the original image. This comparison would not be involved in experiments to determine the shape of the CSF. Some high-pass details (e.g., edges) are noticeably missing from Figure 4.6 (b). This result implies that this CSF model also decays too quickly from the peak to *higher* frequencies when dealing with real images, as opposed to experimental sine waves.

In order to apply the CSF filter to yield a more desirable result, we can adjust the viewing distance parameters. Using values of $d_n = 3h$ and $d_f = \infty$ results in a low-pass filter response and will yield images without perceptual distortion. Figure 4.7 depicts the Goldhill image filtered with the adjusted viewing distance parameters. Note in Figure 4.7 (c) that the low-pass signal is unaffected after filtering.

In a sense, we are no longer using a near and far viewing distance, but rather a "minimum" viewing distance. Viewing the image at less than this distance may cause some distortion to be evident. The main purpose of the CSF filter is to remove perceptually irrelevant details. Recall, however, that quantization is done on a sub-band basis, in which higher spatial frequency bands are coarsely quantized, and therefore there is some functionality overlap between the CSF filter and the quantizer.



(a) Original Goldhill

(b) Goldhill after CSF Filtering

(c) Power Spectra of (a) and (b)

Figure 4.7: Applying the CSF filtering operation with $d_n = 3h$ and $d_f = \infty$.

Another experiment was performed with the CSF filtering component to discover if its incorporation into the WP/HVS codec is desirable. All test images were compressed and reconstructed with and without CSF filtering and the results were assessed using average Q/R ratios of the three quantitative quality measures.[4] Table 4.2 presents the results. Again, the superior results are shown in bold.

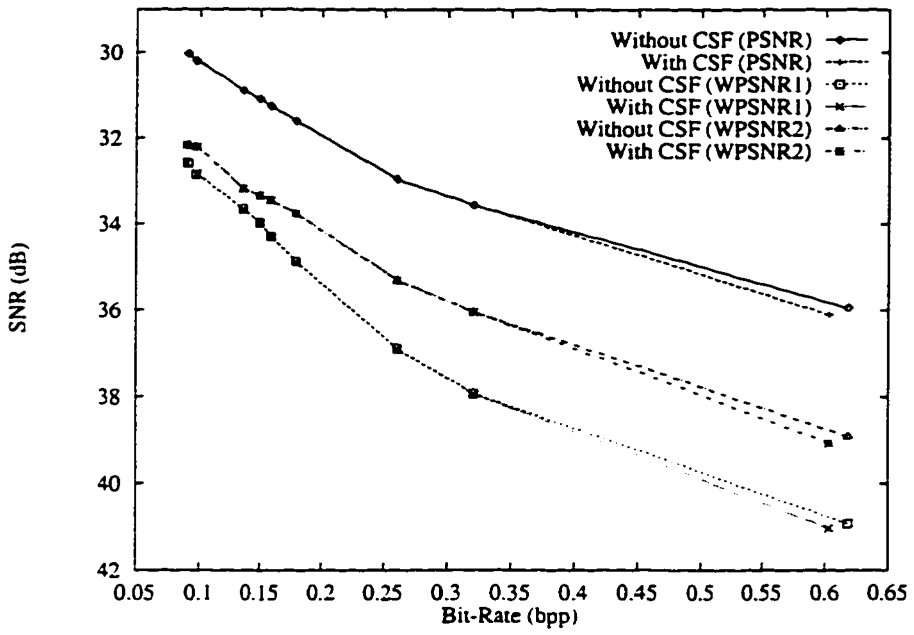
Image		With CSF Filtering		
Category	Bit-Rate	PSNR/R	WPSNR$_1$/R	WPSNR$_2$/R
Natural	0.585	57.05	79.17	**79.11**
Random Texture	1.527	14.81	**24.81**	**28.45**
Structured Texture	0.934	**31.21**	**47.55**	**51.54**
Fingerprint	0.830	**41.88**	**54.42**	**57.04**
Ultrasound	0.606	**58.96**	**67.35**	**65.90**
		Without CSF Filtering		
	Bit-Rate	PSNR/R	WPSNR$_1$/R	WPSNR$_2$/R
Natural	0.593	**61.80**	**80.00**	78.79
Random Texture	1.776	**15.28**	23.24	25.14
Structured Texture	1.054	29.79	41.96	43.76
Fingerprint	0.946	39.29	50.29	52.89
Ultrasound	0.620	57.46	65.72	64.21

Table 4.2: Experimental results from compressing images with and without CSF filter pre-processing, using $d_n = 3h$ and $d_f = \infty$.

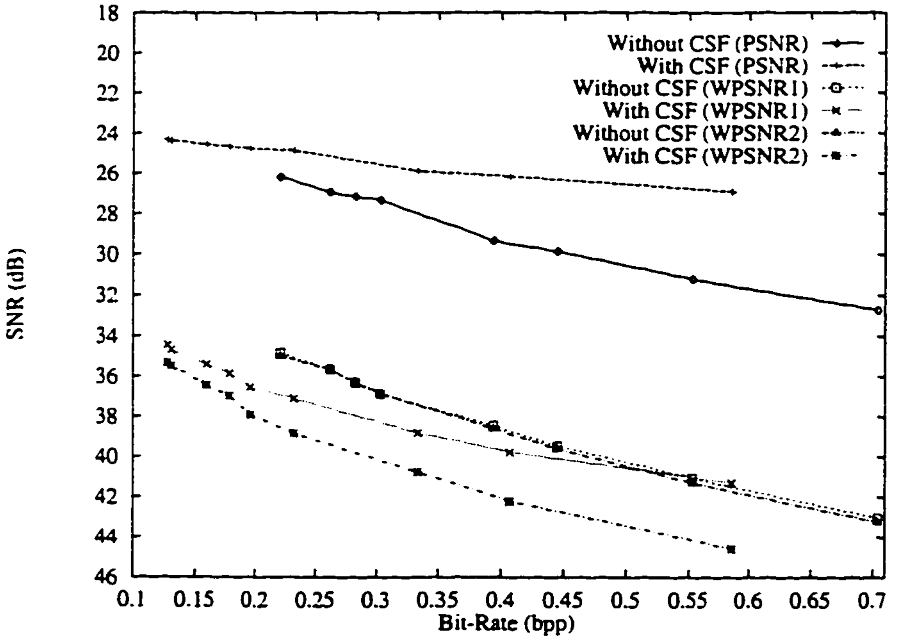
On average, it is clear that incorporating the CSF filtering into the WP/HVS framework can increase the quantitatively judged image quality in most cases. However, it was found that applying the CSF filter did not significantly affect compression results for images with little high-pass energy (e.g., the Ultrasound images). Applying the CSF filter while compressing images with more details and edges (e.g., the Natural image Barbara) did improve the results, at least according to the weighted PSNRs. The standard PSNR measure often decreases with the application of CSF filtering, due to increased distortion. The improvement in weighted PSNRs should be expected, since we introduce distortion only into high-frequency signals. The WPSNR$_1$ and WPSNR$_2$ measures put less weight on high-frequency distortion. Rate-distortion curves for Ultrasound05 and Barbara for the three quality measures are shown in Figure 4.8.

We conclude that CSF filtering *should* be incorporated into the final WP/HVS codec. Results on images with many high-frequency signals can be improved.

[4] To achieve similar bit-rates with and without CSF filtering, the ADF was set to 1.5 when compressing the Natural, Random Texture, and Structured Texture images without CSF pre-processing.



(a) Compression results on Ultrasound05



(b) Compression results on Barbara

Figure 4.8: Comparing the compression performance with and without CSF filtering. Rate-distortion curves (using PSNR, $WPSNR_1$, and $WPSNR_2$) are shown for (a) an image lacking high-frequency information and (b) an image with significant high-frequency information.

4.5 Wavelet Decomposition Component Testing

Once the input image has passed through the HVS pre-processor, the next step is to decorrelate the image using the wavelet packet transform. Recall that we have developed four possible stopping criteria for the packet decomposition (see Section 3.4.2). We need to examine the efficacy of the four different stopping criteria, and the appropriateness of using wavelet packets versus strict pyramidal decomposition. Most researchers have found that wavelet packets do not perform significantly better than pyramidal transform, except perhaps on images with unusual power spectra [65]. However, it is possible that combining wavelet packets with HVS modeling will yield different results.

An experiment was designed to independently test the wavelet packet decomposition component. No gamma correction or CSF filtering was done on the input images. The maximum depth of decomposition was set to $\lambda_{\max} = 4$. Setting $\lambda_{\max} > 4$ yields a large number of sub-bands, takes more computation time, and increases the overhead required for the quantizer parameters. For example, setting $\lambda_{\max} = 5$ would decompose the image to an unreasonable maximum of $4^5 = 1024$ sub-bands. Overhead of this magnitude would prevent a coding gain, since the overhead increases more quickly than wavelet decorrelation efficiency.

The five categories of images (Natural, Random Texture, Structured Texture, Fingerprint, and Ultrasound) were used, giving *54* test images. Each image was compressed using *five* possible wavelet decomposition schemes. These schemes are:

- wavelet packets, *entropy* stopping criterion;
- wavelet packets, *insignificance* stopping criterion;
- wavelet packets, *kurtosis* stopping criterion;
- wavelet packets, *energy* stopping criterion;
- and the standard wavelet transform, *pyramidal* decomposition.

The quality of the reconstructed image was then assessed by the *three* quantitative measures: PSNR, WPSNR_1, and WPSNR_2. The experiment therefore yields $54 \times 5 \times 3 = 810$ different results. All images were compressed using the acceptable distortion factor ADF = 1.0 to enable grouping and comparing results. The wavelet decomposition step will not be affected by different acceptable distortion factors (ADFs) so this restriction is justified.

The wavelet decomposition time, the number of sub-bands, the bit-rate, and Q/R ratios from the three quality assessments were recorded for each combination of image and decomposition scheme. Results from this experiment are shown in Table 4.3. Note the table shows the results averaged over image category. Again, the maximum Q/R ratios are shown in bold.

Image Type	Stopping Criterion	Time (s)	Band Count	Bit-Rate	Quality/Bit-Rate Ratios		
					PSNR/R	WPSNR$_1$/R	WPSNR$_2$/R
Natural	Entropy	9.21	75.8	0.980	39.41	50.61	50.15
	Insignificance	6.80	16.3	0.845	**44.96**	58.76	58.55
	Kurtosis	12.14	35.0	0.944	41.12	53.38	53.14
	Energy	7.80	32.0	0.924	41.72	54.03	53.73
	Pyramid	3.86	13.0	0.846	44.88	**58.84**	**58.68**
Random Texture	Entropy	9.32	128.5	2.600	11.80	17.59	19.27
	Insignificance	6.79	20.9	2.287	**12.75**	19.29	21.05
	Kurtosis	11.82	23.5	2.403	12.29	18.68	20.51
	Energy	7.05	16.4	2.344	12.54	19.01	20.80
	Pyramid	3.87	13.0	2.288	**12.75**	**19.36**	**21.19**
Structured Texture	Entropy	8.02	58.4	1.491	22.40	31.52	33.25
	Insignificance	7.04	27.3	1.423	**23.23**	**32.79**	**34.54**
	Kurtosis	11.39	21.6	1.470	22.70	32.13	33.93
	Energy	7.15	23.5	1.445	22.97	32.43	34.14
	Pyramid	3.89	13.0	1.445	22.85	32.42	34.23
Fingerprint	Entropy	11.37	183.7	1.072	36.85	45.43	48.02
	Insignificance	6.93	81.1	0.918	**40.83**	**52.08**	**54.65**
	Kurtosis	11.63	49.6	0.969	38.67	49.17	51.91
	Energy	9.41	105.1	1.075	36.56	45.00	47.42
	Pyramid	3.88	80.6	0.946	39.29	50.29	52.89
Ultrasound	Entropy	2.33	185.8	0.734	51.61	57.15	56.47
	Insignificance	1.46	32.2	0.611	**57.91**	**65.94**	**64.59**
	Kurtosis	2.71	49.0	0.730	52.28	57.58	57.11
	Energy	1.89	85.3	0.728	52.41	57.83	57.40
	Pyramid	0.82	13.0	0.620	57.46	65.72	64.21

Table 4.3: Average compression results for different image categories and wavelet decomposition schemes.

It is clear that the pyramid decomposition scheme takes the least computation, followed by wavelet packets using the insignificance stopping criterion. Applying the entropy stopping criterion yields a large sub-band count, since the entropy decreases with almost every decomposition. The overhead of the extra sub-bands is obvious from the increased bit-rate of this scheme compared to the others.

Before we further analyze the results, we direct our attention to those who may be critical of the broad image categories, the Natural type in particular. It is important to emphasize that we are only testing the wavelet decomposition schemes, which are mainly affected by the distribution of power in the frequency spectrum. Most natural images have been shown to exhibit a smoothly decaying power spectrum. A brief experiment to investigate the appropriateness of the image categories was executed. The variability of decompositions of images within the same category was examined using the wavelet packet decomposition with the insignificance stopping criterion. The results are shown in Table 4.4.

Image Category	No. of Images	No. of Different Decompositions
Natural	18	5
Random Texture	8	5
Structured Texture	8	4
Fingerprint	10	8
Ultrasound	10	6

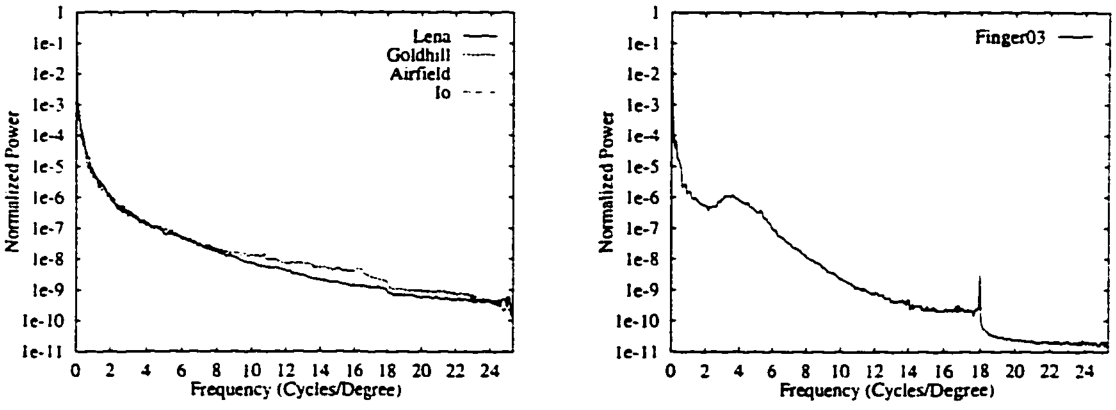
Table 4.4: Variability of wavelet packet decompositions in different image categories.

It was found that Natural images produced relatively few different decompositions. Intuitively, it is correct to group the fingerprint images into the same category, and yet 8 unique decompositions were produced from 10 images. This result is probably due to the varying rotation and depth of field in the FBI fingerprint images. We found that the 8 different decompositions for the fingerprints are "similar" to each other, in that each image exhibits extra decompositions in the sub-bands in levels $\lambda = 2, 3$ and orientations $\theta = 2, 4$, i.e., the horizontal- or vertically-oriented signals.

Continuing our analysis, from a first glance at Table 4.3 it would appear that using wavelet packets with the insignificance stopping criterion is the best decomposition scheme on Structured Texture, Fingerprint, and Ultrasound images by a small margin. The Q/R results are mixed for Natural and Random Texture images. In all cases, the entropy, kurtosis, and energy stopping criteria perform worse than the other two methods.

It is interesting to note that using the insignificance stopping criteria leads to better performance (even according to PSNR) when applied to Fingerprint or Structured Texture images. These types of images have unique power spectra, quite different from that of most Natural images. Using a flexible wavelet decomposition can take advantage of the unusual distribution of energy in the wavelet sub-bands.

Figure 4.9 depicts the 1D power spectra of four Natural images and compares these with one Fingerprint spectrum. Note the bump and spike in the Fingerprint spectrum and the relatively smooth decay of all four Natural images.



(a) (b)

Figure 4.9: Example power spectra of (a) four Natural images and (b) a Fingerprint image.

A simple statistical test based on the T statistic was applied to discover whether there was a significant difference between the results from the wavelet decompositions using pyramid or the insignificance measure. The test revealed that there was no statistical difference between the two methods at an alpha level of 0.1. However, it may be more helpful to further test these two decomposition schemes, subjectively and objectively measuring the compression results of individual images. We shall examine both in more detail in the subsequent chapter when comparing the WP/HVS codec to other codecs.

4.6 Quantization Component Testing

Once the image has been decomposed into sub-bands by the wavelet transform, the resulting coefficients will be quantized by the WP/HVS codec. The framework supports both uniform and non-uniform quantization (see Section 3.5). Experiments were performed in order to determine which quantizer should be applied in general applications. No HVS pre-processing was done and only pyramidal wavelet decomposition was used during the experiments.[5]

4.6.1 Uniform Quantization Component

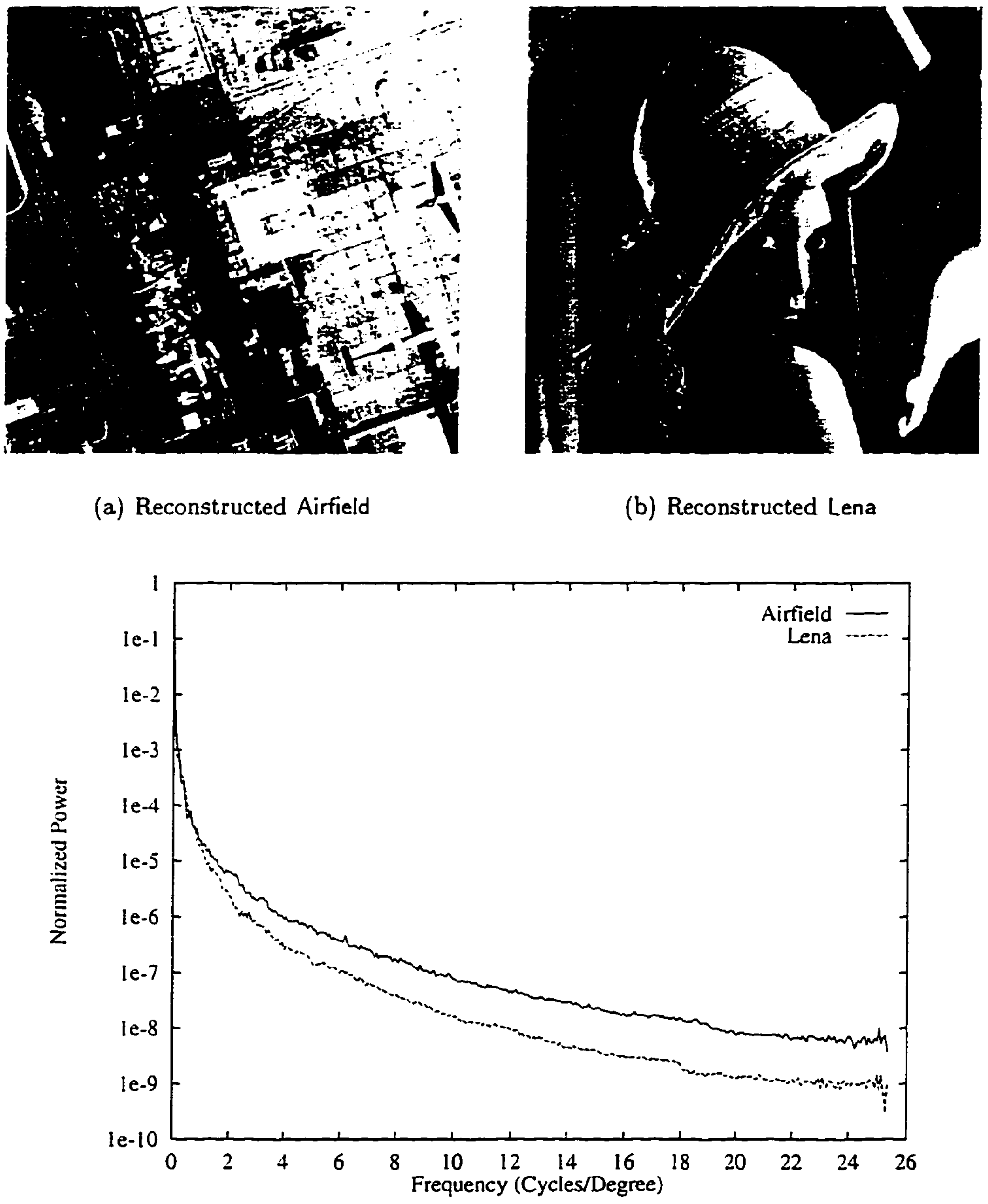
The uniform quantizer sets each coefficient whose magnitude is below an insignificance threshold c_I to zero. All 54 test images were compressed using the default highest perceptual quality setting (i.e., ADF = 1.0). The quantizer stop-band is therefore at its narrowest. The percentage of coefficients set to 0 by the stop-band was recorded. The average results are shown in Table 4.5 along with the time required for quantization and dequantization.

Image Type	Quantization Time (s)	Dequantization Time (s)	Total Time (s)	$c < c_I$ (%)
Natural	2.62	2.23	4.85	83.4
Structured Texture	2.87	2.54	5.41	67.9
Random Texture	3.43	3.17	6.60	46.7
Fingerprint	2.42	2.08	4.51	80.6
Ultrasound	0.56	0.48	5.06	85.4

Table 4.5: Time required and percentage of wavelet coefficients set to zero using uniform quantization.

There is a trend to the percentage of insignificant coefficients. If an image has lower average energy, a larger percentage of wavelet coefficients will have small magnitude. Many of these small coefficients will be below the insignificance threshold c_I and can be set to zero. This phenomenon is demonstrated in Figure 4.10. The Airfield image resulted in the lowest percentage (73.9%) of zero coefficients out of all Natural test images. The well-known Lena image resulted in the highest percentage (90.2%) of

[5] Using wavelet packet decomposition with the insignificance stopping criterion was also carried out with the two different quantization schemes. This decomposition did not appear to modify the results significantly from using pyramidal decomposition, so we only report the pyramid results here for simplicity.



(a) Reconstructed Airfield

(b) Reconstructed Lena

(c) Power Spectra of (a) and (b)

Figure 4.10: The (c) power spectra of (a) Airfield reconstructed with 73.9% of wavelet coefficients set to zero and (b) Lena reconstructed with 90.2% of wavelet coefficients set to zero.

zero coefficients over all categories.[6] As can be seen from Figure 4.10 (c), Lena has less energy on average throughout most of the frequency spectrum. Therefore, it is easier to compress Lena without introducing perceptible error. In fact, to yield similar quality, Figure 4.10 (a) is compressed to 1.29 bpp (PSNR = 32.12) and Figure 4.10 (b) to 0.53 bpp (PSNR = 36.24). In this case, Lena can be compressed more *and* yield a higher quality rating.

Recall from Section 3.5.3 that the acceptable distortion factor (ADF) can be set to higher than 1.0 to facilitate greater compression. Results with different ADFs will be discussed in the next chapter, in which we compare the WP/HVS codec with other codecs.

4.6.2 Non-Uniform Quantization Component Testing

The non-uniform quantizer in the WP/HVS framework was designed with the purpose of modeling the HVS masking effect. More distortion can be accepted in areas of intense signal content, which translates to wavelet coefficients of larger magnitude. Figure 4.11 shows quantization errors taken from level $\lambda = 3$, orientation $\theta = 2$ of the Barbara image. The number of quantizer bins was 63 for uniform quantization and 17 for non-uniform quantization. As can be seen by this figure, the masking effect is being modeled to a degree, since the non-uniform quantizer results in larger errors for larger coefficients.

As mentioned in Section 3.5.4, non-uniform quantization has a more complex quantization algorithm and a less complex dequantization algorithm than uniform quantization. Table 4.6 gives empirical evidence for this complexity difference. Non-

Image Type	Quantization Time (s)	Dequantization Time (s)	Total Time (s)	Time Increase Over Uniform (%)
Natural	3.15	1.98	5.12	5.6
Structured Texture	3.89	2.31	6.20	14.6
Random Texture	5.08	2.93	8.01	21.5
Fingerprint	3.15	1.91	5.06	12.2
Ultrasound	0.65	0.44	1.09	5.5

Table 4.6: Time required for non-uniform compared to uniform quantization.

[6] The fact that over 90% of the wavelet coefficients can be set to zero without perceptual image distortion suggests that Lena is quite *easy* to compress. This fact is perhaps more significant than first assumed due to the incredibly prevalent use of the Lena image in reporting codec results in the literature.

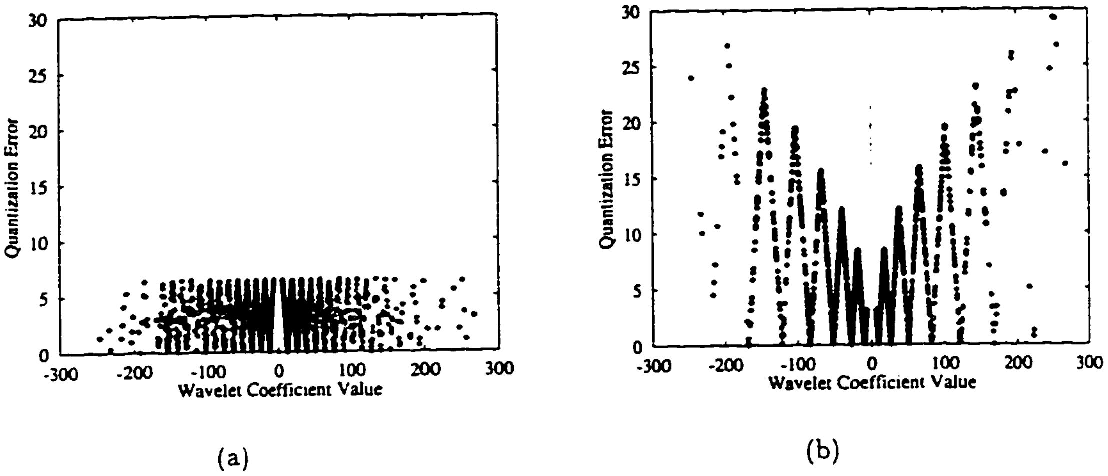


Figure 4.11: Absolute quantization error resulting from (a) the uniform quantizer with 63 bins and (b) the non-uniform quantizer with 17 bins applied to a sub-band of wavelet coefficients from Barbara.

uniform quantization requires more total computation. The small increase in computation will only be worthwhile if non-uniform quantization can produce consistently higher quality results than uniform.

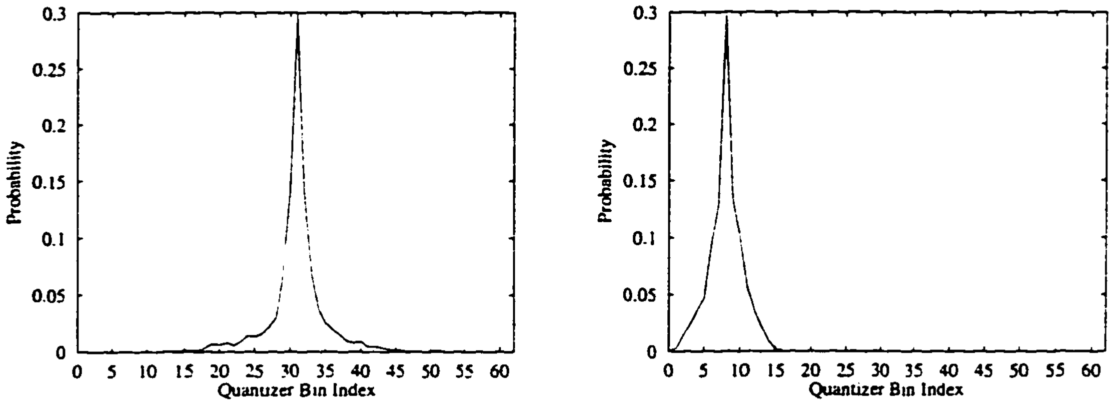
All test images were compressed using uniform and then non-uniform quantization. Table 4.7 shows the average Q/R ratios for each test image category.[7] From this table, it is clear that the uniform quantizer will usually outperform the non-uniform quantizer in terms of Q/R ratios.

	Quality with		**Quality with**	
Image	**Uniform Quantization**		**Non-Uniform Quantization**	
Type	WPSNR$_1$/R	WPSNR$_2$/R	WPSNR$_1$/R	WPSNR$_2$/R
Natural	**58.84**	**58.68**	56.95	55.69
Structured Texture	**32.42**	**34.23**	31.17	32.26
Random Texture	**19.36**	**21.19**	19.04	20.44
Fingerprint	**50.29**	**52.89**	46.03	47.44
Ultrasound	**65.72**	**64.21**	51.51	51.15

Table 4.7: Comparing the compression performance using uniform versus non-uniform quantization.

[7] The PSNR/R quality measure is not shown for simplicity, but its results are similar.

Why does applying the non-uniform quantizer not improve the results? We suggest it is primarily due to interactions with the entropy coding procedure. A brief experiment was executed to compare the input to the arithmetic coder from both the uniform quantizer and the non-uniform quantizer. This input is simply the quantizer bin indices. Using a sub-band from the Barbara image, the uniform quantizer had 63 levels and the non-uniform had 17. The bin indices from the uniform quantizer had an entropy of 3.58 compared to 3.17 from the non-uniform quantizer. The average absolute quantization error increases from 2.28 per coefficient with the uniform quantizer to 4.85 per coefficient with the non-uniform. Figure 4.12 depicts the normalized histograms of the output of both types of quantizers. Both histograms are plotted using the same x-axis scale to allow direct comparison.



(a) (b)

Figure 4.12: Distribution of quantization symbols (i.e., bin indices) from (a) the uniform quantizer and (b) the non-uniform quantizer when applied to a sub-band of wavelet coefficients from Barbara.

Notice how similar the distributions are in shape and in entropy, even though the uniform quantizer has 46 more possible bins. This empirical evidence suggests that the small decrease in entropy from the non-uniform quantizer cannot make up for the larger increase in error. It appears unlikely that the non-uniform quantizer could beat the uniform quantizer in the trade-off between entropy and error.

Our experiments with non-uniform quantization support the research results found by others, that the increased complexity incurred from incorporating non-uniform quantization does not yield improvements in image quality over simple uniform quantization in practical applications [57, 5, 2, 64, 65]. Additionally, we have examined the interactions between the non-uniform quantizer and the adaptive arithmetic coder. and have confirmed that an optimally designed uniform quantizer coupled with an entropy coder will perform similarly to a non-uniform quantizer [28].

From the analysis of our experimental results, we conclude that non-uniform quantization should not be used in the WP/HVS codec.

4.7 Summary

The experiments conducted in this chapter have enabled us to draw conclusions about the WP/HVS framework components. We can use this information to outline the structure of the proposed WP/HVS codec.

1. Gamma correction will not be used. It may be important to use when the hardware does not implicitly do gamma correction, but this is a rare case.

2. The CSF filter can be applied using near distance $d_n = 3h$ and far distance $d_f = \infty$. The filter will only have a significant impact on images with high-energy high-pass spectra.

3. The only stopping criterion kept for the wavelet packet transform is the coefficient insignificance. Entropy, energy, and kurtosis have been eliminated. The standard pyramidal decomposition scheme will also be kept as an option in the codec.

4. The non-uniform quantizer has been eliminated. Only uniform quantization will be used in the WP/HVS codec.

Now that some WP/HVS framework components have been discarded, the resulting WP/HVS codec configuration can be tested and its results compared with other codecs in the next chapter.

Chapter 5

Performance Analysis of the WP/HVS Codec

5.1 Introduction

The previous chapter has stream-lined the WP/HVS framework to two main configurations. Each configuration uses contrast sensitivity function (CSF) filtering and uniform quantization to reduce high-frequency signals to which the human visual system (HVS) is less sensitive. The two final configurations differ only in the wavelet decomposition stage. in which either wavelet packets with the insignificance stopping criterion or standard pyramidal decomposition can be applied. We will henceforth call the two codecs WP/HVS Insignificance and WP/HVS Pyramid. WP/HVS codec will refer to either configuration.

This chapter examines the results of compressing several representative images with the WP/HVS codec and two other techniques. Presented herein are:

- a quantitative performance analysis using rate-distortion curves;
- a qualitative performance analysis using original and reconstructed images and distortion maps;
- and a brief treatment of codec complexity differences.

5.2 Images and Codecs Involved in the Comparison

The image compression performance of the WP/HVS codec may now be compared to techniques developed by others. In so doing, we will use several representative images taken from our experimental image library (see Section 4.2.1). The Lena and

Barbara images are Natural images with insignificant and significant amounts of high-frequency details, respectively. We will also test the Natural image Airfield during the qualitative analysis. The two texture images shown in Figure 4.2, Texture08 and Texturel5, will be used as examples of Random Texture and Structured Texture images. Finally, the Fingerprint and Ultrasound images shown in Figure 4.3 will also be used for comparisons.

We have chosen two other codecs to compare with the WP/HVS codec results. The wavelet compression algorithm using set partitioning in hierarchical trees (SPIHT) developed by Said and Pearlman has been shown to perform strongly when results are assessed using PSNR [51]. This algorithm uses the wavelet transform but does not attempt to model any aspects of the HVS, so it will be instructive to compare with the WP/HVS codec. SPIHT incorporates embedded trees of zero-magnitude wavelet coefficients, a quantization technique the WP/HVS codec does not yet support. The results of the standard JPEG algorithm will also be examined since it is still heavily used in applications and performs well when compressing for high quality at medium-high bit-rates [57]. The JPEG standard uses a quantization matrix which is based on HVS sensitivity (see Section 2.4.1).

5.3 Quantitative Comparison Analysis

We have stressed that qualitative analysis of compression results is the most revealing quality assessment technique, since the human observer is the final judge of any reconstructed image. However, a quantitative comparison of the three codecs (WP/HVS, SPIHT, and JPEG) is not subjective and therefore offers repeatable, theoretical proof of results.

To quantitatively compare the performance of the codecs, we compressed the sample images to several different bit-rates and recorded the resulting PSNR and WPSNR_1 values. We have chosen to be conservative and have not used the WPSNR_2 measure in this experiment. WPSNR_2 is designed using the same filtering mechanism as our CSF filter, and the critical reader might suspect that WPSNR_2 could be optimized in favor of the WP/HVS codec. The WPSNR_1 measure is also somewhat of a standard based on a quality analysis factor of the television industry (see Section 3.6.2).

It should be emphasized that the weighted PSNR measurements are more reliable quality assessors than the standard PSNR, since they are designed to better correlate with human judgment. The WPSNRs are limited in that they do not attempt to model the HVS masking effect, but they do take into consideration the variable HVS sensitivity to distortion of different spatial frequencies. PSNR does not explicitly model any HVS characteristics.

The WP/HVS codec was set to compress the sample images to nine different bit-rates by setting the acceptable distortion factor (ADF) to values in the interval $[1, 5]$ in increments of 0.5. Each resulting bit-rate from the WP/HVS Pyramid codec was then recorded. The SPIHT codec was then used to compress and reconstruct the images at the identical bit-rates. The bit-rate cannot be set with the JPEG codec, so a trial-and-error method was used, setting the JPEG quality factor to different levels to attempt to mimic the bit-rates of the WP/HVS codec.

The bit-rate versus PSNR and WPSNR_1 results were then plotted on two separate graphs for each test image. These graphs are shown in Figures 5.1 through 5.6. Before discussing the individual rate-distortion curves, we will comment on some noticeable patterns in the results.

- The SPIHT codec is designed to be optimized for achieving maximal PSNR, since it encodes the largest wavelet coefficients first. Not surprisingly, SPIHT performs the best on all but Structured Texture and Fingerprint images, according to PSNR. Once again, PSNR is not a good quality assessor, so optimizing for it is a questionable goal.

- On the PSNR rate-distortion graphs, the WP/HVS codec reaches a maximum PSNR (as the bit-rate increases) more quickly than the other codecs. That is, the WP/HVS curves level-off faster than those for SPIHT and JPEG. This phenomenon is caused by the contrast sensitivity function (CSF) pre-processing stage, which effectively sets a threshold for the PSNR before any activity by the uniform quantizer. This trend does not occur on the WPSNR_1 charts since the weighted PSNR measure takes into account the HVS spatial frequency sensitivity.

- The WP/HVS Insignificance codec yields superior performance on Structured Texture and Fingerprint images, even beating SPIHT according to PSNR. This significant performance increase is due to the flexible wavelet packet decomposition scheme used in the WP/HVS Insignificance codec. Incorporating wavelet packets allows our codec to be *adaptable* to image content, rather than forcing a pyramidal decomposition. Our top-down implementation of the packet decomposition can provide better performance on images with unusual power spectra with little increase in computation. On images with more typical power distributions, the WP/HVS Insignificance codec does not incur significant performance loss (compared to SPIHT) and in most cases it wins according to the weighted PSNR, which is a better quality assessor.

We now discuss specific details of the results from each test image, referring to the Figure in which the graphs are subsequently presented.

Lena—**Figure 5.1.** The Lena image is a Natural image which is fairly smooth, that is, it has little energy in its high-frequency power spectra. SPIHT consistently beats all other codecs when measured by PSNR. However, it is difficult to visually discern the differences between the two WP/HVS codecs and SPIHT, and this result is supported by the $\mathrm{WPSNR}_\mathrm{l}$ measure.

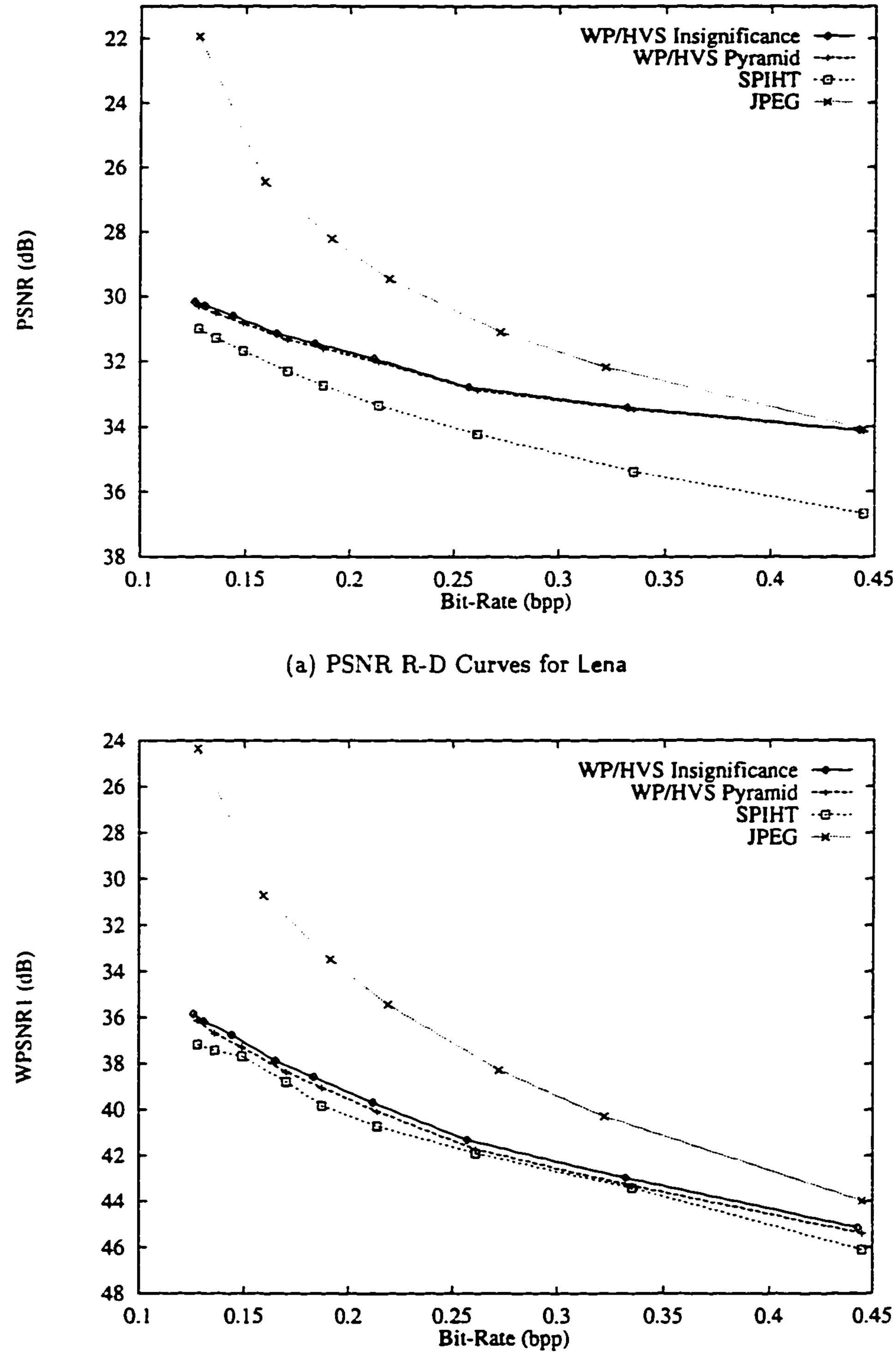
Barbara—**Figure 5.2.** Barbara has much more energy in high frequencies than Lena. The WP/HVS codecs can exploit this different energy distribution, since subbands of higher frequencies are quantized coarsely due to the wider HVS stopband. Again, SPIHT is shown to be superior to all other methods from a PSNR point of view. As suggested earlier, JPEG does well at compressing this image at higher bit-rates. However, the more important $\mathrm{WPSNR}_\mathrm{l}$ measure shows the WP/HVS codecs to beat SPIHT at bit-rates below 0.4 bpp. There does not appear to be any significant difference in results between the WP/HVS codec using packets or pyramidal decomposition.

Texture08—**Figure 5.3.** The Random Texture image Texture08 yields results similar to that of Barbara, most likely due to its preponderance of high-frequency signals. In this case, note that WP/HVS and SPIHT produce similar PSNR at rates below 0.6 bpp. It is also interesting to see that JPEG again shows its strengths at higher bit-rates when it beats all other methods at 1.7 bpp according to $\mathrm{WPSNR}_\mathrm{l}$.

Texture15—**Figure 5.4.** Recall from Chapter 4 that using wavelet packet decomposition with the insignificance stopping criterion appeared to perform better than straight pyramidal decomposition on Structured Texture images. The compression results of Texture15 support that conclusion, from both PSNR and $\mathrm{WPSNR}_\mathrm{l}$ measurements. This method is only beaten by SPIHT at high bit-rates according to PSNR.

Finger03—**Figure 5.5.** The Fingerprint images were another image category in which wavelet packet decomposition appeared appropriate to use. Applying the WP/HVS codec with packet decomposition using the insignificance stopping criterion beats all other codecs at every bit-rate according to PSNR and WPSNR_1. Our experimental results provide evidence that it is correct that wavelet packets are particularly successful when applied to images with unusual power spectra, like Structured Texture and Fingerprint images.

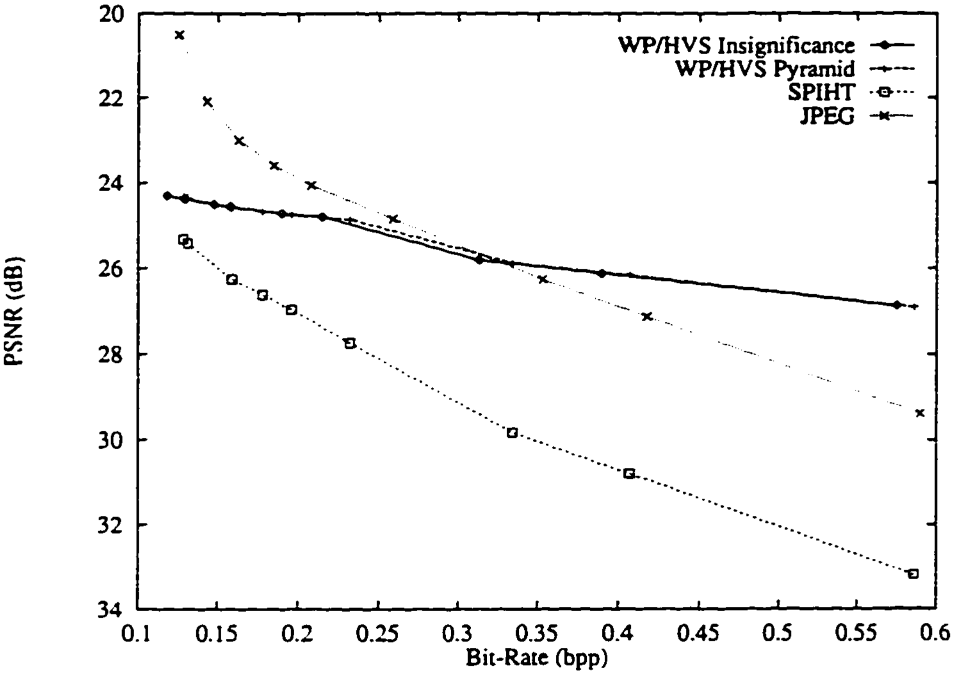
Ultrasound05—**Figure 5.6.** The Ultrasound05 image has even less high-frequency information than the Lena image discussed above. The results of the three wavelet-based codecs do not appear to be significantly different from one another with this image based on PSNR or WPSNR_1. The JPEG method appears to yield much worse results.



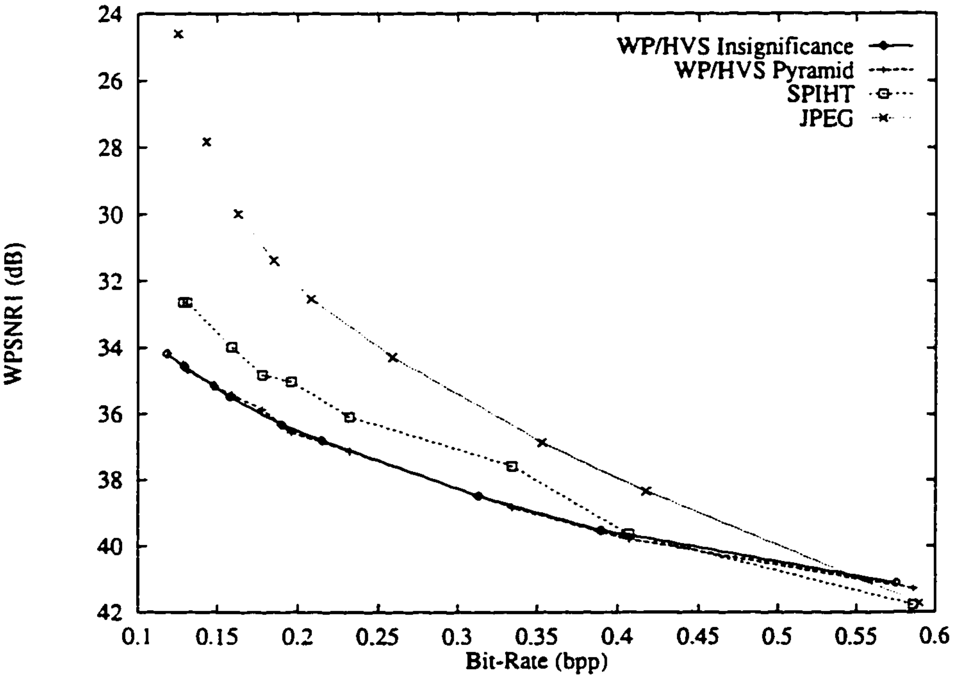
(a) PSNR R-D Curves for Lena

(b) WPSNR_1 R-D Curves for Lena

Figure 5.1: Comparing the quantitatively assessed compression results of the four codecs on the Lena image using (a) PSNR and (b) WPSNR_1.

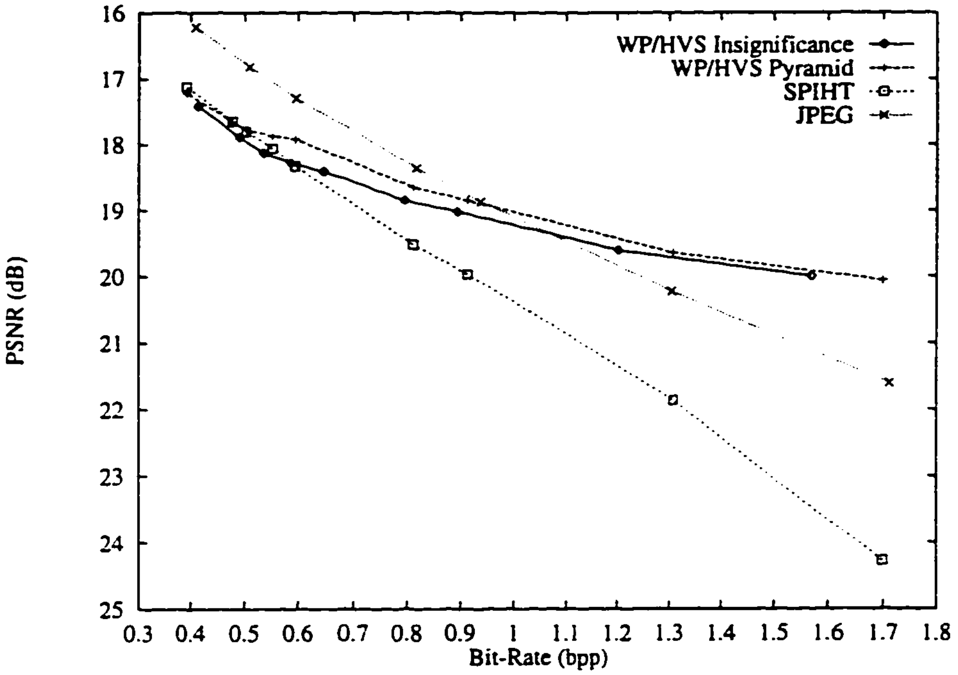


(a) PSNR R-D Curves for Barbara

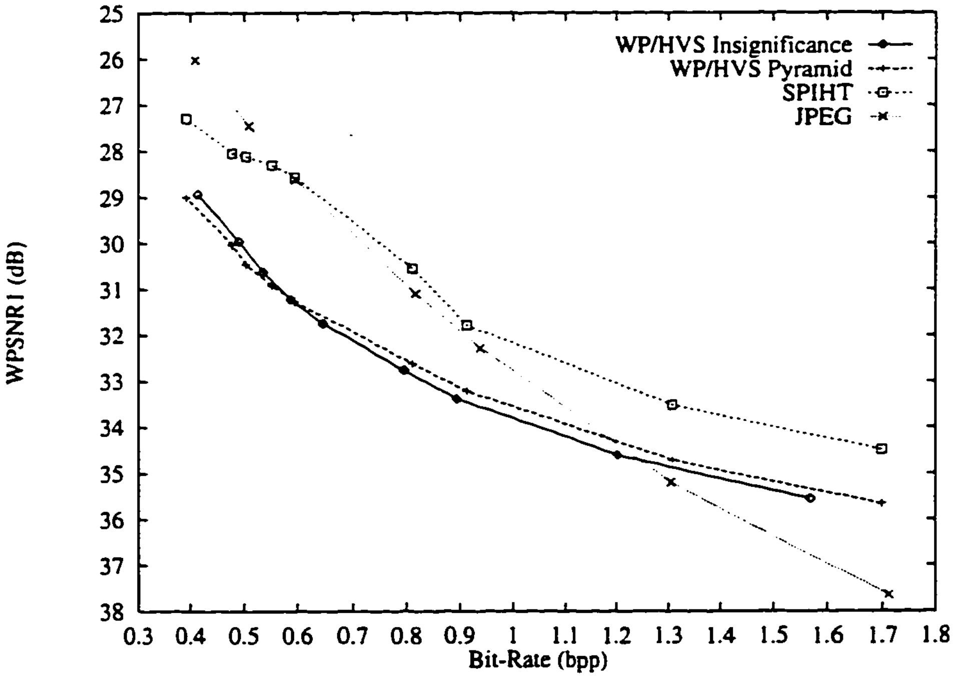


(b) WPSNR$_1$ R-D Curves for Barbara

Figure 5.2: Comparing the quantitatively assessed compression results of the four codecs on the Barbara image using (a) PSNR and (b) WPSNR$_1$.

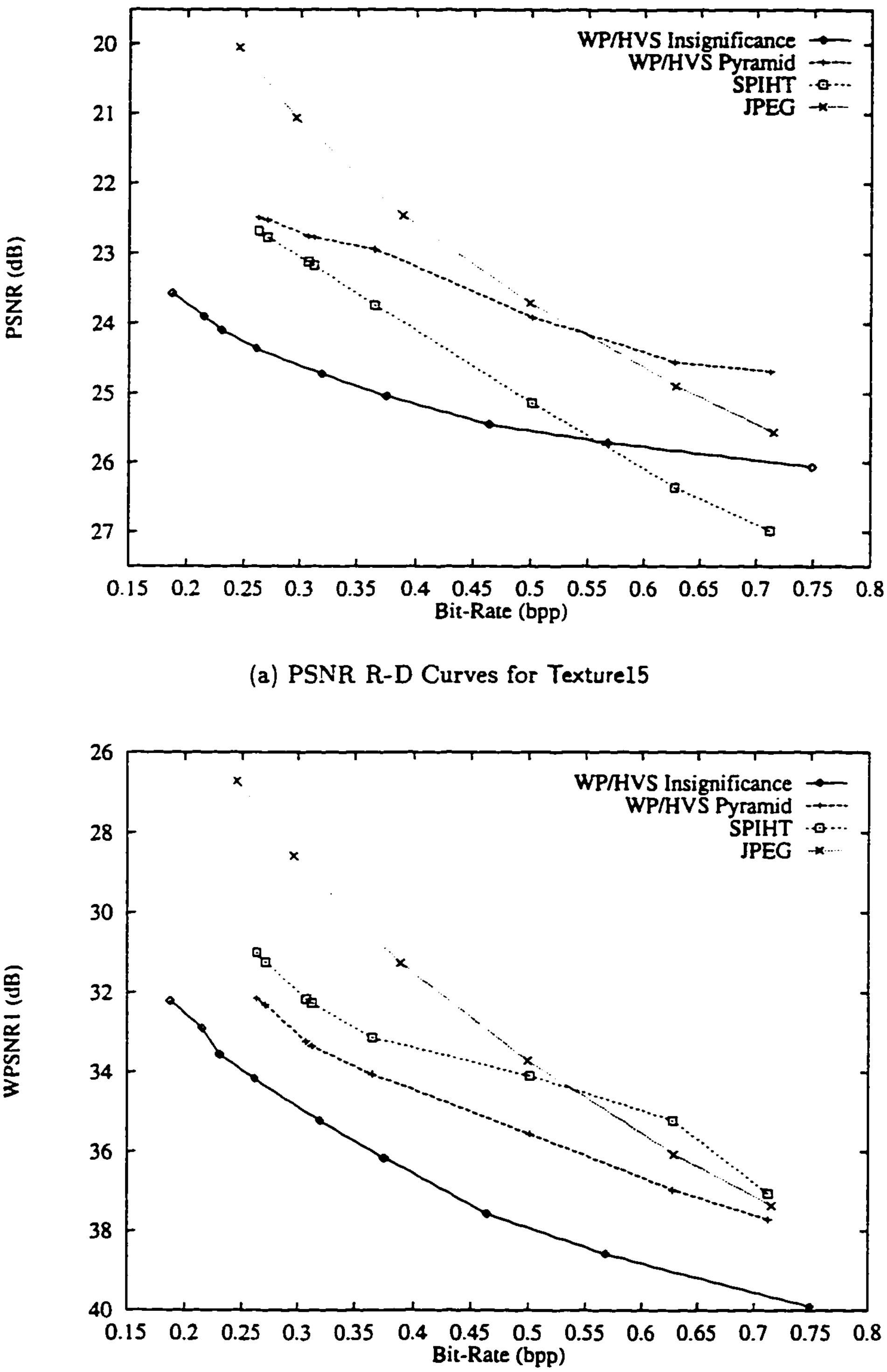


(a) PSNR R-D Curves for Texture08



(b) $WPSNR_1$ R-D Curves for Texture08

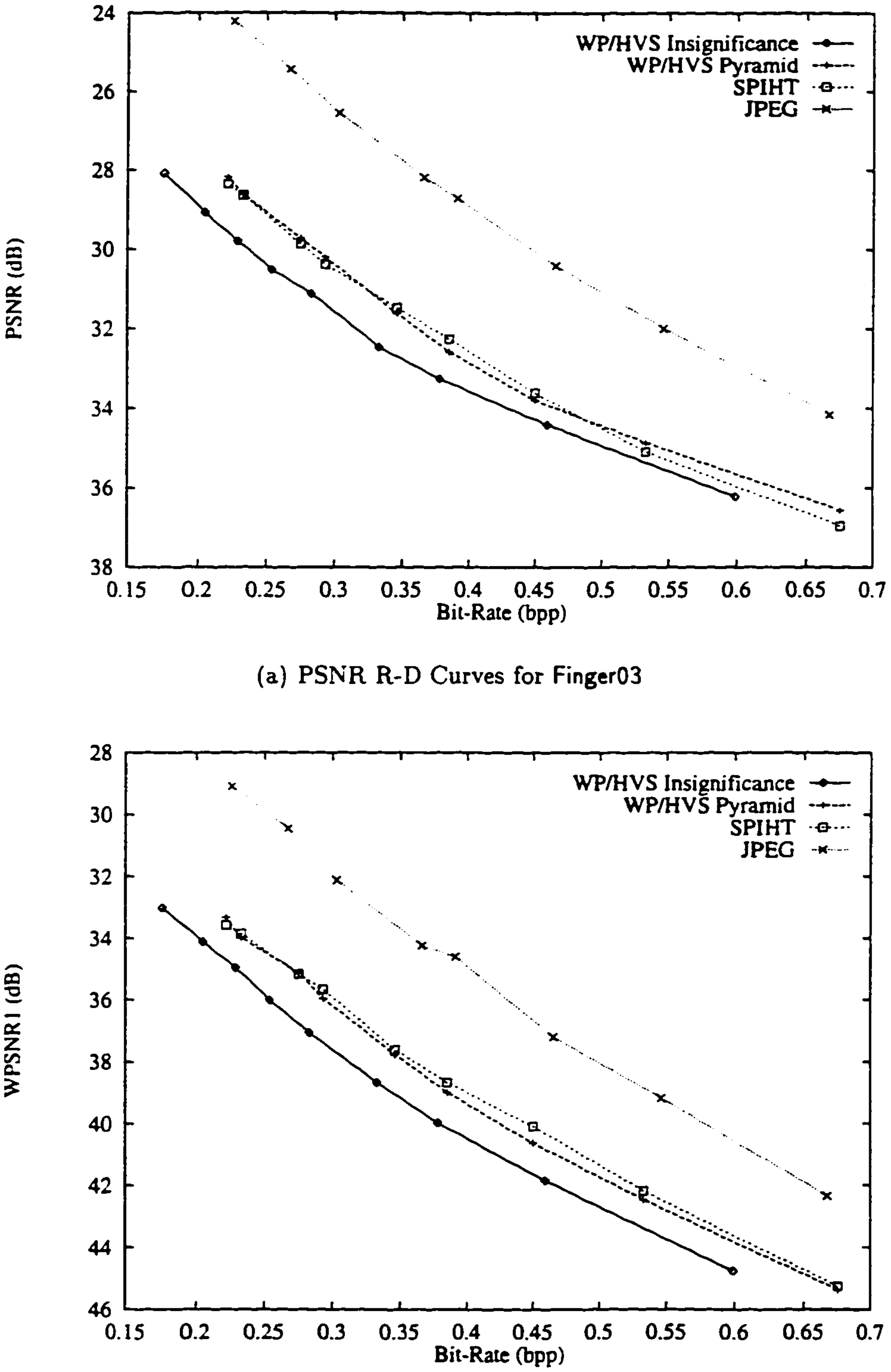
Figure 5.3: Comparing the quantitatively assessed compression results of the four codecs on the Texture08 image using (a) PSNR and (b) $WPSNR_1$.



(a) PSNR R-D Curves for Texture15

(b) WPSNR$_1$ R-D Curves for Texture15

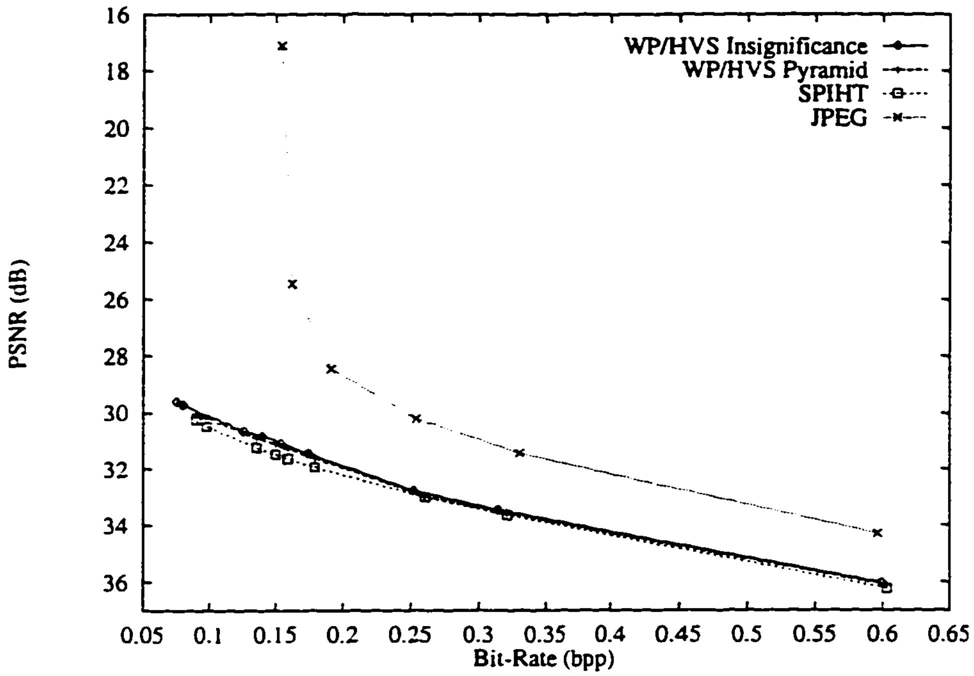
Figure 5.4: Comparing the quantitatively assessed compression results of the four codecs on the Texture15 image using (a) PSNR and (b) WPSNR$_1$.



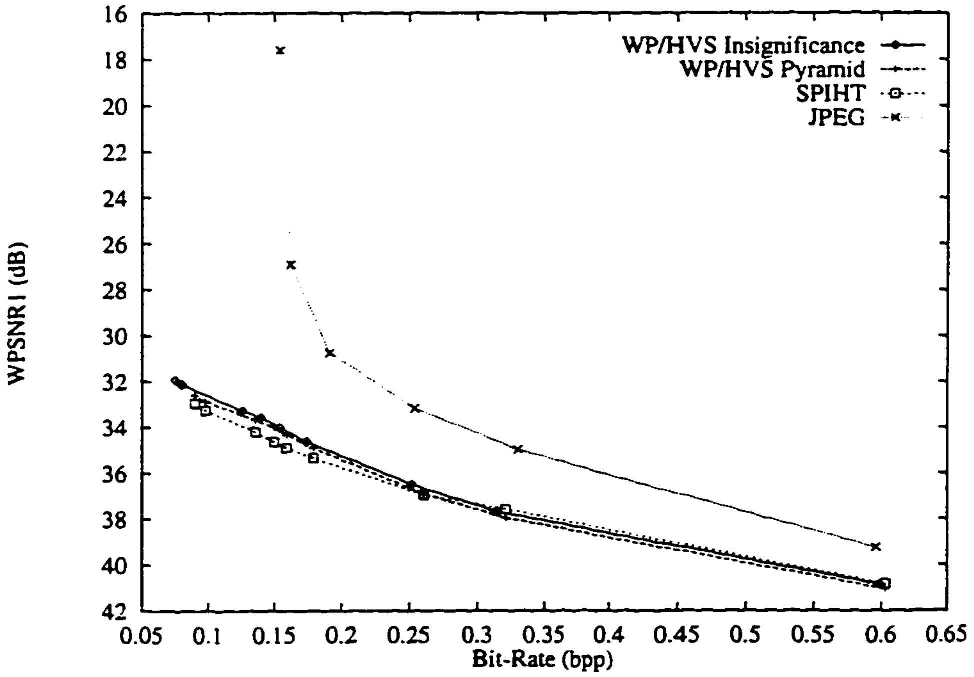
(a) PSNR R-D Curves for Finger03

(b) WPSNR_1 R-D Curves for Finger03

Figure 5.5: Comparing the quantitatively assessed compression results of the four codecs on the Finger03 image using (a) PSNR and (b) WPSNR_1.



(a) PSNR R-D Curves for Ultrasound05



(b) $WPSNR_1$ R-D Curves for Ultrasound05

Figure 5.6: Comparing the quantitatively assessed compression results of the four codecs on the Ultrasound05 image using (a) PSNR and (b) $WPSNR_1$.

5.4 Qualitative Comparison Analysis

We stress that the final recipient, that is the user, of the reconstructed image after compression is the most important quality assessor. It is therefore prudent to subjectively examine some results from the various codecs. For this stage of analysis we have dropped JPEG from the comparison. This decision was made for two main reasons. The first is for simplicity, since presenting the subjective results requires a great deal of space. The second reason is that JPEG produces significant blocking artifacts at low-medium bit-rates. The quantitative quality assessors do not reflect the "blockiness", but subjectively this distortion is annoying.

Distortion maps (produced using a viewing distance of four times the picture height) will be shown with the squared error in each pixel. Note that the maps depict the greatest distortion in black. That is, darker pixels represent more distortion. Our subjective judgments of image quality have been carried out by viewing the results on a high quality, 17" color monitor. Some details of results will be unavoidably changed in the process of printing the images on a laser printer using dithering.

We have chosen to look at the test images from the previous section at a "medium" bit-rate for each image, that is, we use ADF = 2.5 for each image. This ADF introduces enough distortion to be visible to facilitate visually examining the reconstructed images and distortion maps. Using this ADF allows each image to be compressed without introducing unacceptably annoying distortion. In other words, this is probably a usable ADF. However, due to the printer dithering problem, we increase the ADF to 5.0 for **Structured Texture** and **Fingerprint** images to emphasize differences in results.

The reconstructed Lena images from SPIHT and the WP/HVS Pyramid codec are shown in Figure 5.7. The results show that the two images are difficult to visually distinguish. Again, this phenomenon is most likely due to the lack of significant edges and high-frequency details in Lena.

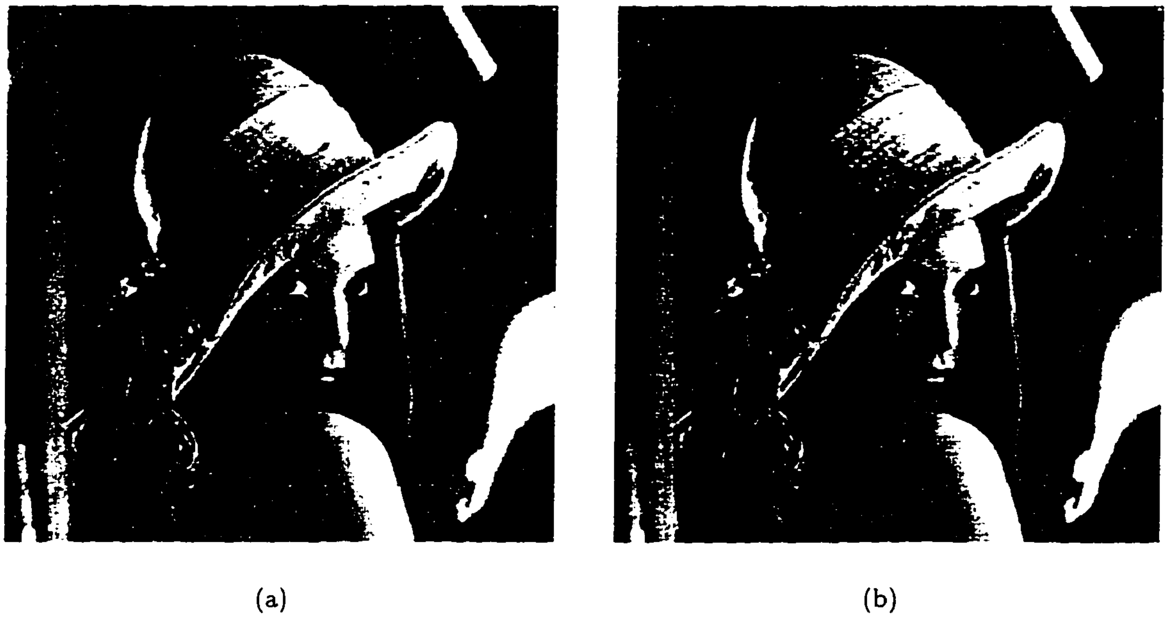
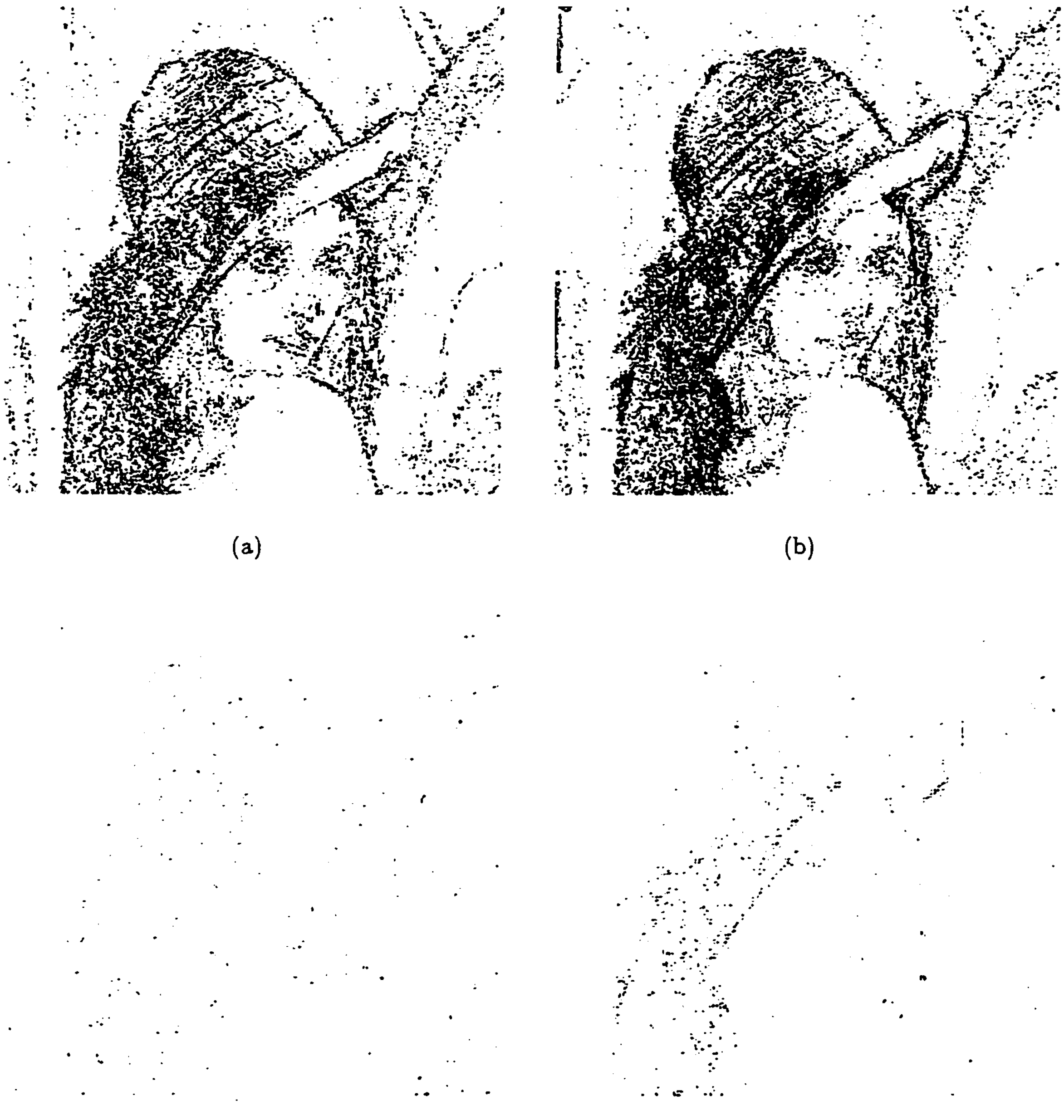


Figure 5.7: Reconstructed Lena image at 0.214 bpp from (a) SPIHT and (b) WP/HVS Pyramid.

We present all four distortion maps for Lena in Figure 5.8. Upon close inspection, especially of the WPSNR_1 distortion maps, it is evident that the WP/HVS codec introduces more distortion in the feather in Lena's hat. However, SPIHT reconstructs the image with greater distortion around the mouth and nose. Recall that distortion is more visible to the HVS in smooth image areas, so concentrating the distortion in the "busy" region of the feather like the WP/HVS does is desirable.



(a)

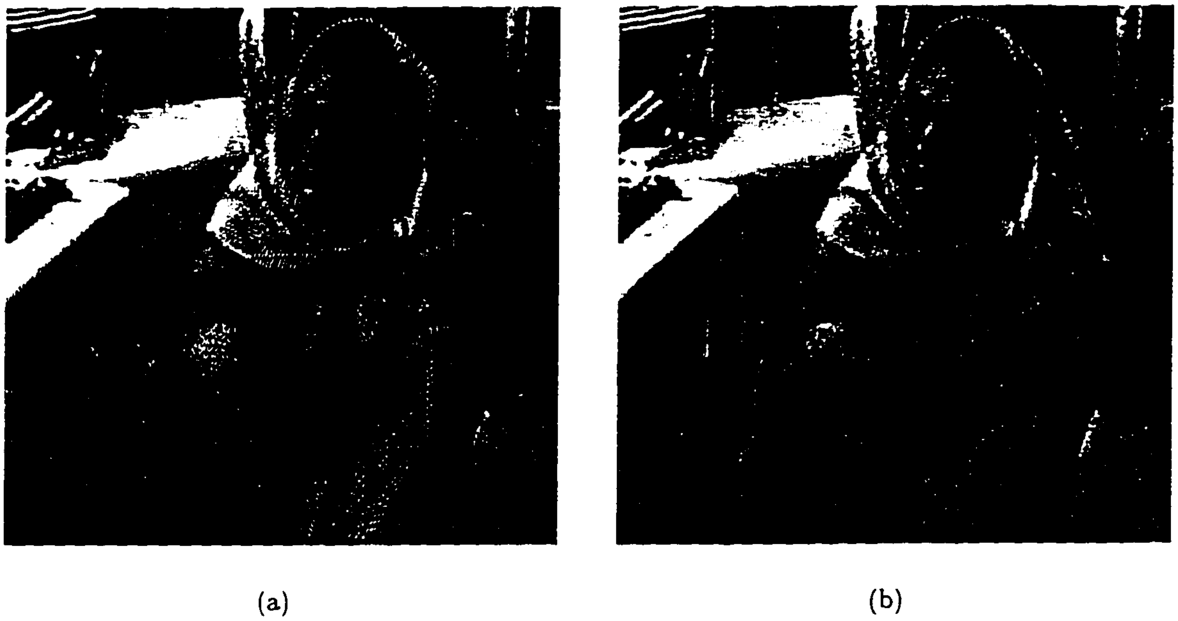
(b)

(c)

(d)

Figure 5.8: Distortion maps from the reconstructed Lena image at 0.232 bpp: (a) PSNR of SPIHT result; (b) PSNR of WP/HVS Pyramid result; (c) WPSNR_1 of SPIHT result; and (d) WPSNR_1 of WP/HVS Pyramid result.

The WP/HVS codec was shown to achieve lower PSNR than SPIHT with the Natural image Barbara. Figure 5.9 depicts the reconstructed images for each technique.

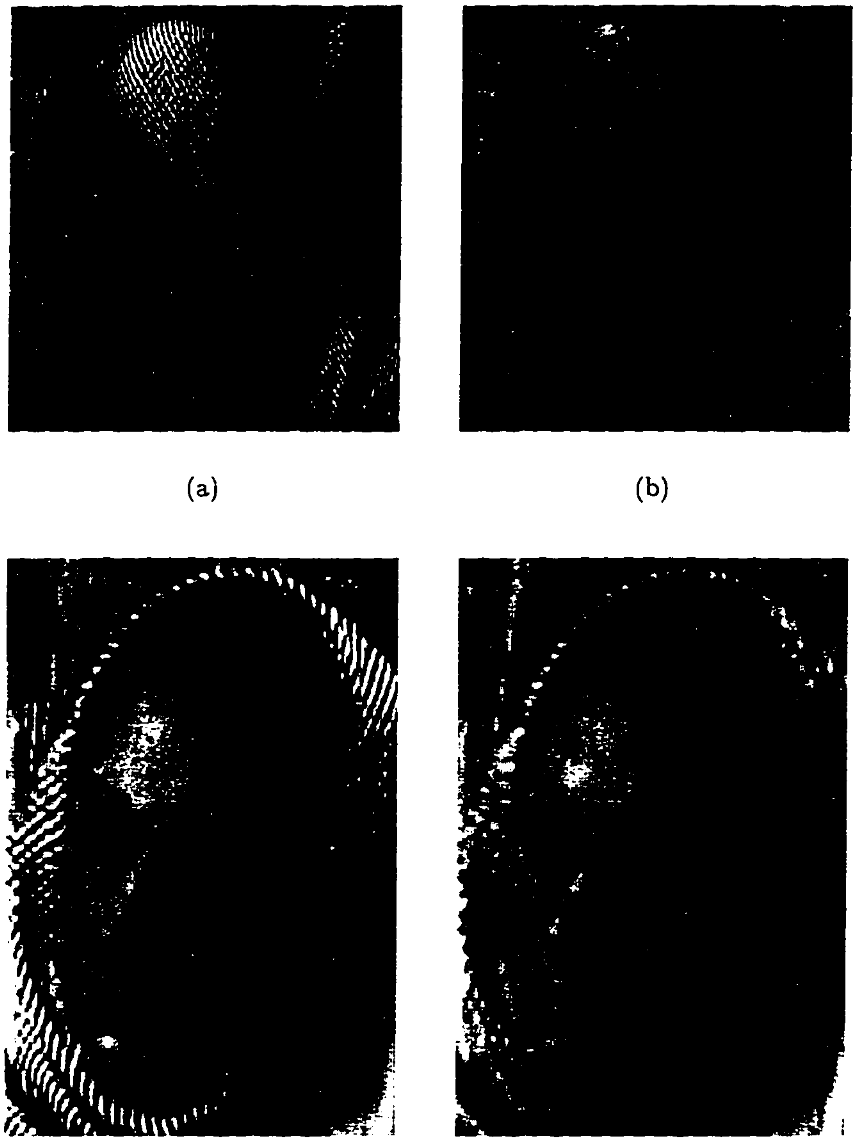


(a)

(b)

Figure 5.9: Reconstructed Barbara image at 0.232 bpp from (a) SPIHT and (b) WP/HVS Pyramid.

Notice in the cropped images in Figure 5.10 that some high-frequency details in the pants and scarf have been eliminated from the WP/HVS result. However, the pants are artificially detailed and "noisy" using SPIHT. The WP/HVS codec is able to reconstruct the image with less distortion in the face, arms, and table cloth.



(a)

(b)

(c)

(d)

Figure 5.10: Cropped regions from the reconstructed Barbara image at 0.232 bpp from (a, c) SPIHT and (b, d) WP/HVS Pyramid.

Figure 5.11 shows the distortion maps from the WPSNR_l measure. The lower distortion at the edges of the arms and in the face is particularly noticeable in the WP/HVS codec WPSNR_l map. The edges in an image are perhaps the most important constituent, since boundaries define the shape and size of objects. The fact that the WP/HVS can preserve edge detail is a strong benefit over SPIHT. Additionally, image processing techniques exploring comprehension and understanding often rely on strong edge information. If these techniques are applied to compressed images, it is important to have clear edges.

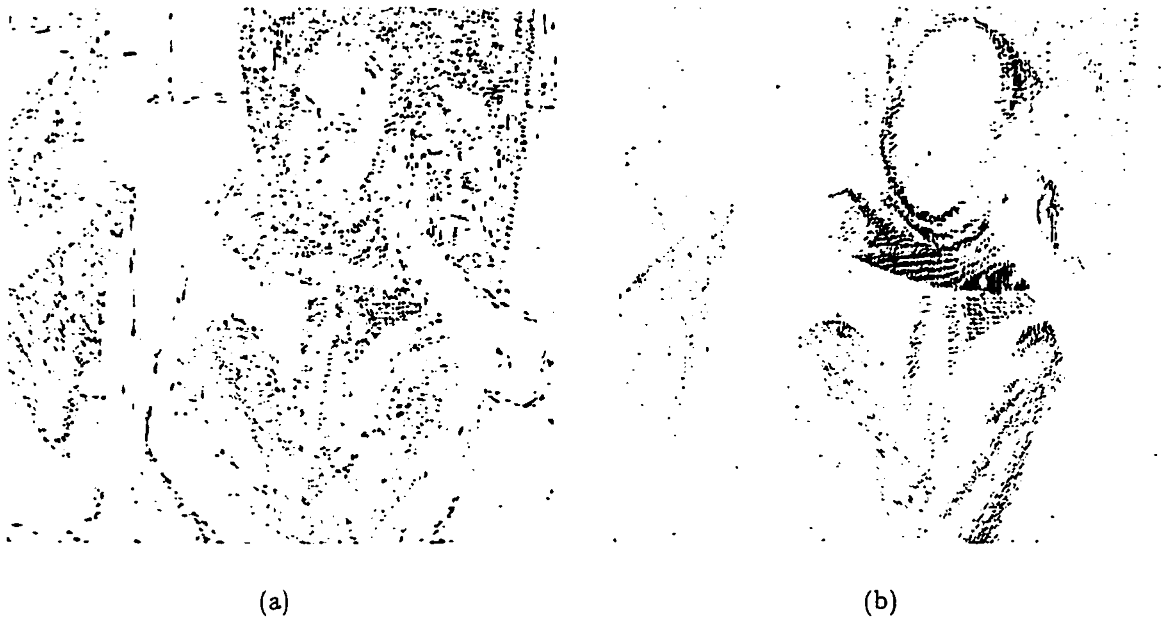


Figure 5.11: Distortion maps from the reconstructed Barbara image at 0.232 bpp: (a) WPSNR_l of SPIHT result and (b) WPSNR_l of WP/HVS Pyramid result.

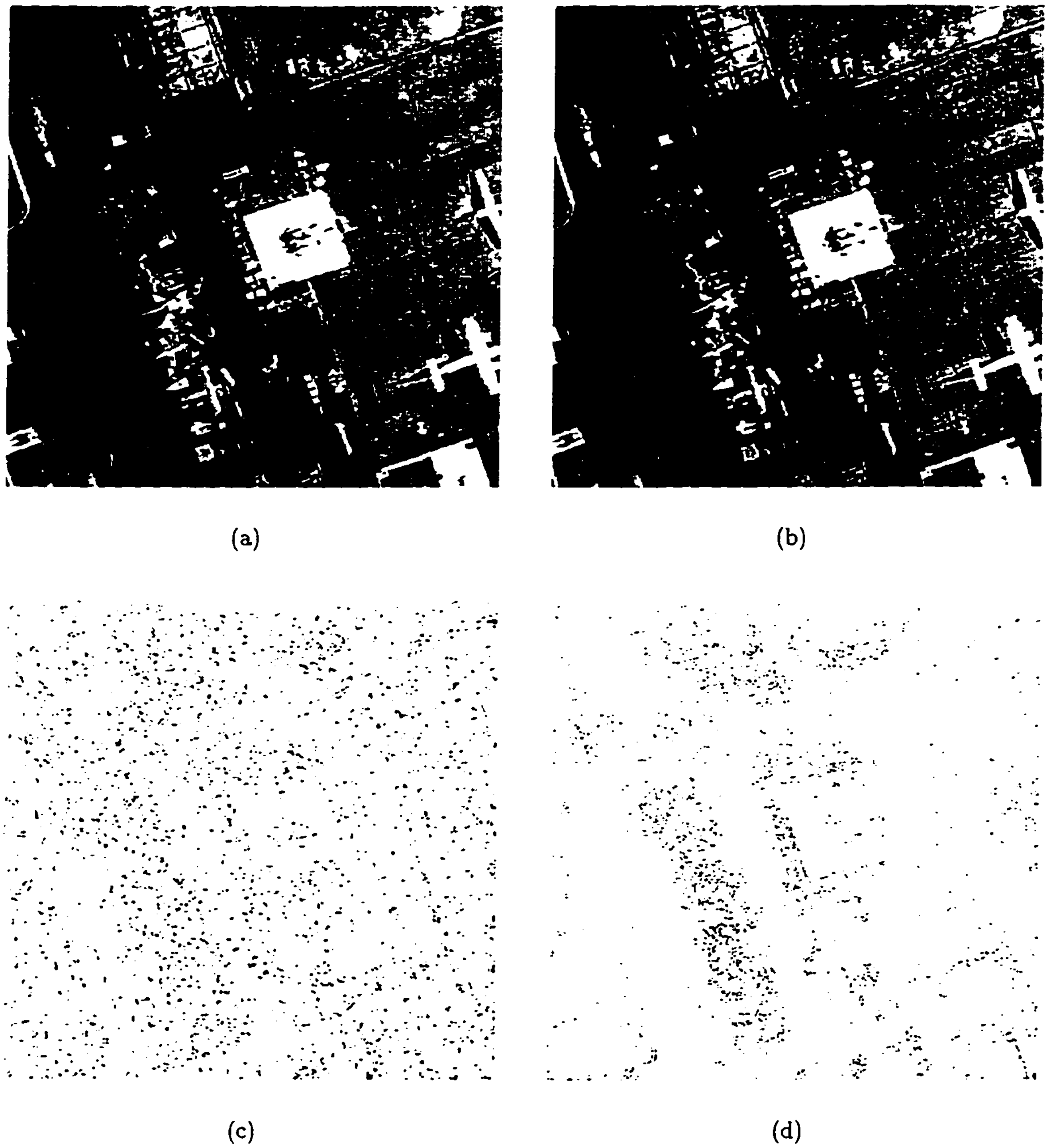
Ideally, distortion should be just below the visible threshold at each location in the image. Note in the Barbara image that this threshold should vary throughout the image due to the structures present. In order to take advantage of the HVS masking effect, distortion should be placed in a high-frequency region, like Barbara's scarf. Figure 5.11 (b) shows that the distortion in the scarf under the face is in a similar frequency and orientation to the original scarf signal and should therefore be masked to the HVS. This effect demonstrates that the WP/HVS codec *can* model the masking effect implicitly through the use of CSF pre-processing, which removes some high-frequency details. Therefore, more distortion will be placed in these areas.

Figure 5.12 presents the compression results for the Natural image Airfield.[1] The figure contains reconstructed images and WPSNR$_1$ distortion maps for Airfield using both SPIHT and WP/HVS Pyramid codecs. Again in this Natural image, it is clear from the distortion maps that the WP/HVS codec has concentrated distortion in the highly active image region containing many airplanes, in the bottom left portion of the image. SPIHT, in contrast, appears to have spread distortion almost equally over the entire image. In an image with distinct objects, this strategy is not in concordance with the HVS sensitivity.

We have shown that applying wavelet packets to Structured Texture images can produce performance gains over pyramidal decomposition. Figure 5.13 shows the reconstructed Texture15 image and distortion maps from the WP/HVS Pyramid and the WP/HVS Insignificance codecs for direct comparison. The pyramid codec introduces more artifacts into the reconstructed image than using packets, even when the pyramid result is compressed at a slightly higher bit-rate. It is evident from Figures 5.13 (c) and (d) that the WP/HVS Pyramid codec introduces more distortion which is distributed evenly throughout the image. This even distribution of errors is in contrast to the Barbara or Airfield images, but is desirable in Texture15 since the visible threshold of distortion would not change drastically throughout the image.

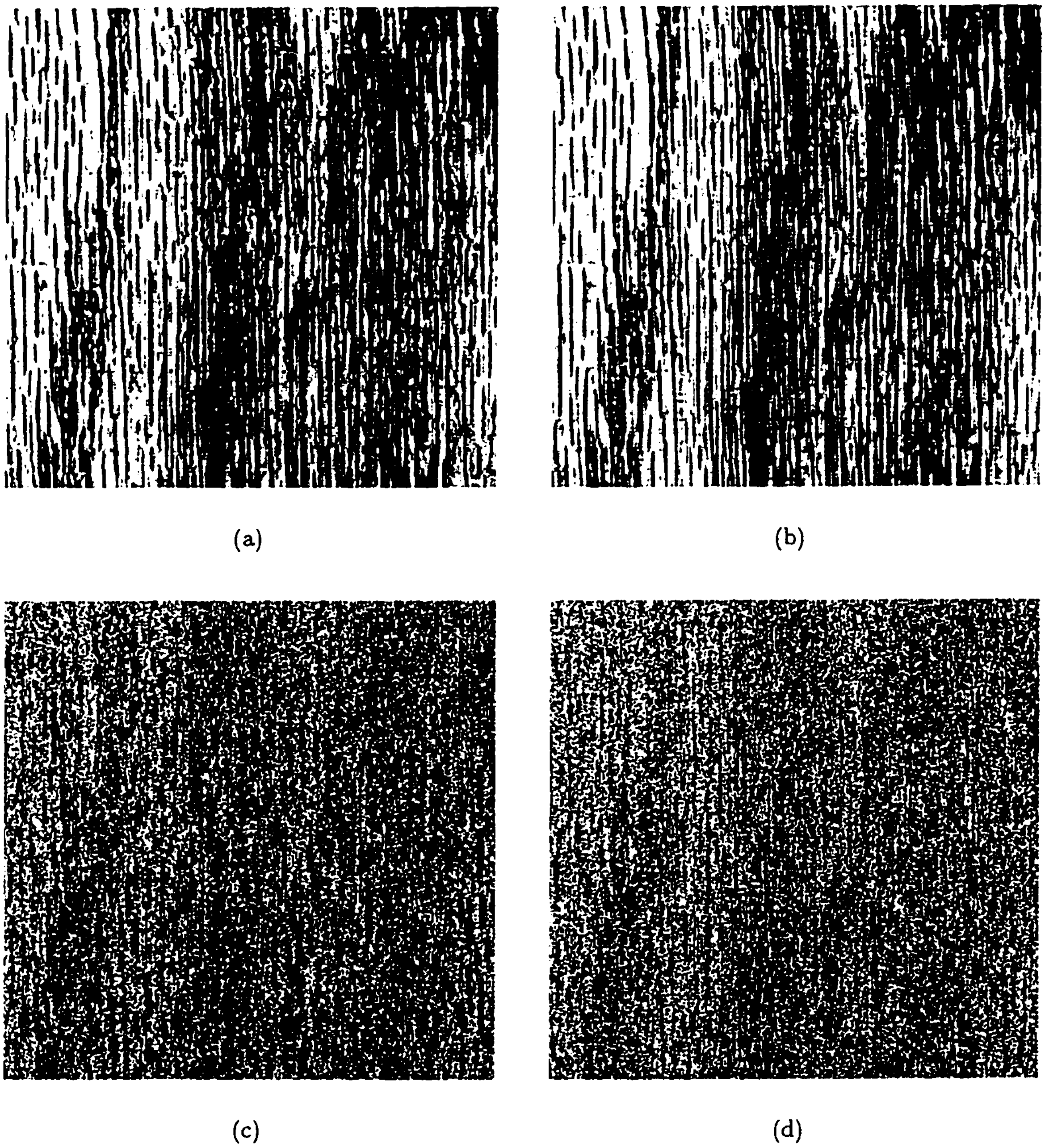
Wavelet packets have also been shown to increase the compression performance over pyramidal decomposition when applied to fingerprint images. Figures 5.14 (a) and (b) depict the reconstructed Finger03 image from the SPIHT and WP/HVS Insignificance codecs. The SPIHT result exhibits substantially more blurring than the WP/HVS Insignificance result, particularly on the left half of the image. The distortion maps in Figures 5.14 (c) and (d) show the SPIHT image to have generally more error distributed throughout the image.

[1] Figures 5.12 through 5.14 are presented after the discussions of results for the Airfield, Texture15, and Finger03 images.


(a) (b)

(c) (d)

Figure 5.12: Reconstructed images and distortion maps from the Airfield image at 0.415 bpp: (a) reconstructed image from SPIHT; (b) reconstructed image from WP/HVS Pyramid; (c) WPSNR$_l$ of SPIHT result; and (d) WPSNR$_l$ of WP/HVS Pyramid result.



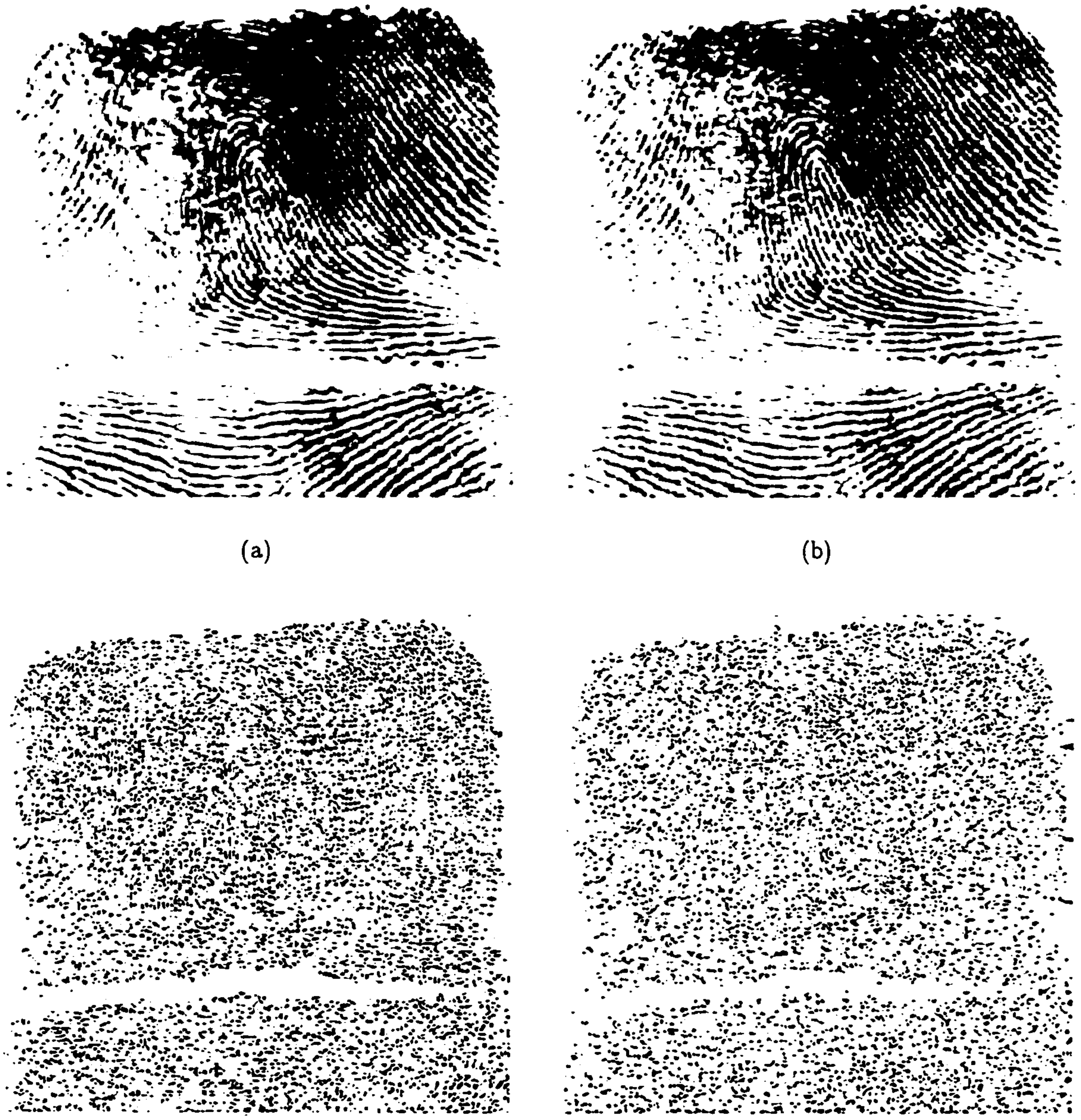
(a)

(b)

(c)

(d)

Figure 5.13: Reconstructed images and distortion maps from the Texture15 image: (a) reconstructed image from WP/HVS Pyramid at 0.195 bpp; (b) reconstructed image from WP/HVS Insignificance at 0.187 bpp; (c) WPSNR_l of WP/HVS Pyramid result; and (d) WPSNR_l of WP/HVS Insignificance result.



(a) (b)

(c) (d)

Figure 5.14: Reconstructed images and distortion maps from the Finger03 image at 0.177 bpp: (a) reconstructed image from SPIHT; (b) reconstructed image from WP/HVS Insignificance; (c) WPSNR$_1$ of SPIHT result; and (d) WPSNR$_1$ of WP/HVS Insignificance result.

5.5 Complexity Comparison

An experiment was conducted to assess the differences in computational time requirements for the JPEG, SPIHT, WP/HVS Pyramid, and WP/HVS Insignificance codecs. The six test images involved in the quantitative analysis (see Section 5.2) were compressed and decompressed using all four codecs. The times required to accomplish these tasks were recorded and averaged and are presented in Table 5.1.

Codec Name	Compression Time (s)	Decompression Time (s)	Total Time (s)
JPEG	0.75	0.43	1.18
SPIHT	2.97	2.72	5.69
WP/HVS Pyramid	22.49	5.25	27.74
WP/HVS Insignificance	25.46	5.93	31.39

Table 5.1: Computational complexity comparison of compression/decompression time for the JPEG, SPIHT, WP/HVS Pyramid, and WP/HVS Insignificance codecs.

There are several points to note about these results. The JPEG code has been optimized by hand by programmers from the Joint Photographic Experts Group. The other codecs are being tested using un-optimized "research code". All four codecs require more time to compress than to decompress the input images. It appears at first glance that the WP/HVS codecs require substantially more time. It is important to note that approximately 75% of the WP/HVS compression time is taken by the CSF filtering process, which requires that the forward and reverse fast Fourier transforms be performed. Even with the CSF filtering, the WP/HVS codec is substantially faster than many second generation approaches [56].

5.6 Summary

This chapter has focused on comparing the performance of two configurations of the WP/HVS framework (i.e., WP/HVS Pyramid and WP/HVS Insignificance) with two other codecs, JPEG and SPIHT. JPEG is a popular compression industry standard and the wavelet-based SPIHT algorithm is well-known from the compression literature to have high performance.

Images of each category were compressed and the results were reported both quantitatively and qualitatively. The WP/HVS Insignificance codec appears to out-perform the other codecs on Structured Texture and Fingerprint images, based on both quantitative and qualitative analysis. On other types of images it is difficult to distinguish any difference in the results from the two WP/HVS codecs. Considering the minimal difference in complexity between the two approaches, it would most likely be safe to always use the WP/HVS Insignificance codec, which may actually end up with a pyramidal decomposition. The flexible wavelet packet decomposition approach is adaptive to image signal content and will therefore not restrict any image to a forced pyramidal decomposition. The adaptivity can take advantage of images with unusual power spectra.

For Natural and Random Texture images, SPIHT out-performs the WP/HVS codec according to PSNR measurements. However, the weighted PSNR measure WPSNR_l indicates that the WP/HVS codec is similar or superior to SPIHT with these images. We again stress that the weighted PSNR measure is designed to correlate more with the HVS and therefore results measured with WPSNR_l are more indicative than those measured with PSNR. In fact, when comparing the WP/HVS codec to SPIHT, PSNR can actually increase with decreasing perceptual quality.

On images with insignificant high-frequency energy, e.g. any Ultrasound image, the results from the various codecs are not visually distinguishable from each other. However, with some images with significant high-frequency energy, e.g. Barbara, the WP/HVS results are subjectively better. The contrast sensitivity function (CSF) pre-processing built into the WP/HVS codec removes some high-frequency image signal details. We found that this stage implicitly allows the WP/HVS codec to model the HVS masking effect, by distributing more distortion in detailed image regions and less distortion in smooth regions. The WP/HVS codec was also found to retain more of the important edge information that the SPIHT technique.

Chapter 6

Conclusions and Future Directions

6.1 Introduction

This final chapter is dedicated to clearly pinpointing the contributions of this thesis to the research area of image compression. We present a brief summary of each chapter, a list of specific contributions. and several ideas for further research.

6.2 Thesis Summary

The following is a brief review of the entire thesis, summarizing each chapter's main discussion points.

Chapter 2. A general overview of image compression and quality assessment techniques was presented. including a model codec (compressor-decompressor). We discussed spatial- and transform-domain mapping techniques which have been used to decorrelate the input image. including a critical review so-called second generation methods. The wavelet transform and wavelet packet generalization were also presented. We discussed the advantages and disadvantages of three different quantization techniques: uniform, non-uniform, and vector. A model of the human visual system (HVS) incorporating sensitivity to luminance, spatial frequency, and image signal content was also presented.

Chapter 3. Building upon the fundamental background information discussed within Chapter 2, Chapter 3 presented the core ideas behind the research of this thesis. The ideas were presented in the context of an image compression framework, supporting HVS pre-processing, several different wavelet decomposition strategies, both uniform and non-uniform quantization, and tools for quality assessment. This system has been named the WP/HVS framework (i.e., image compression using wavelet packets and human visual system characteristics).

Chapter 4. The options and parameters of the WP/HVS framework needed to be restricted in order to design a fully-functioning, general-purpose WP/HVS codec. In Chapter 4, we report on a "greedy" approach to designing the codec by independently examining each framework component. The relatively large, categorized, experimental image library was also introduced. The experiments executed and reported on within this chapter provided the necessary data for designing the WP/HVS codec from the framework.

Chapter 5. Having determined the final two configurations of the WP/HVS codec (i.e., WP/HVS Pyramid and WP/HVS Insignificance), Chapter 5 applied these codecs and compared their performance to two other well-known codecs on several different images of varying type. The other systems used were the JPEG (Joint Photographic Experts Group) and a wavelet-based algorithm using set partitioning in hierarchical trees (SPIHT).

6.3 Thesis Contribution

This thesis has provided several core contributions to the body of research literature on image compression. The following points clearly define the contribution.

1. **Overview.** A critical overview of image compression techniques concentrating on wavelets and incorporating HVS characteristics was provided. The review indicated that more research is needed on wavelet packets and modeling the HVS, and on the possible benefits of applying them in tandem. We also provided justification for *not* concentrating on second generation compression methods by presenting many challenges faced by these approaches.

2. **Wavelet Packets.** We have examined wavelet packets and have developed and compared several stopping criteria for a computationally-light, top-down decomposition approach. An efficient quadtree representation of wavelet decomposition was developed. We have examined the power spectra of different images and image types to determine how and why wavelet packet decomposition yields different results from the pyramidal wavelet transform.

3. **HVS Modeling.** Several critics have suggested that attempts to incorporate HVS characteristics into compression have not been particularly successful thus

far [13]. This point is precisely one of our inspirations for further investigation of HVS modeling. We have examined techniques for modeling the sensitivities to luminance, spatial frequency, and signal content (i.e., the masking effect).

Gamma correction was investigated as one method of modeling luminance sensitivity. It was quickly found to be unnecessary due to the built-in gamma correction in most hardware, which supported previous research.

An anisotropic model of the the contrast sensitivity function (CSF) was used to design a pre-processing filter, which can eliminate some visually less significant details from the input image. Through experiments, we found that the CSF model could only be applied in the WP/HVS codec after some modifications. The changes made force the CSF filter to pass all low-frequency signals. The previously defined CSF model could not be directly applied in the codec, which probably suggests that HVS threshold experiments using sine waves can only approximate the effects of viewing real images.

A quantizer "stop-band" was designed based on HVS frequency sensitivity experiments performed by Watson *et al.*, which sets small-magnitude wavelet coefficients to zero, and also sets the default uniform quantizer bin size [58, 59]. The original Watson experiments were carried out using the wavelet transform and their results could be directly incorporated into our codec. Other techniques presented in the literature use an arbitrarily set global threshold below which wavelet coefficients are set to zero [53, 51]. Our method is based on experimental evidence of the HVS sensitivity to wavelet coefficients and gives a unique threshold to each wavelet transform sub-band. This is extremely important considering the fact that humans are less sensitive to errors in higher frequencies and diagonal orientations.

A non-uniform quantizer was designed to take advantage of the masking effect, that is, the effect that greater distortion is visually acceptable in more active areas of an image. We were able to develop the non-uniform quantizer in a unique way, without requiring companding or an exhaustive trial-and-error method. However, it was found that applying the non-uniform quantizer did not improve compression results. We discussed this result in the context of supporting other researchers' findings and in examining the interaction between the non-uniform quantizer and symbol encoder.

4. **Combining Wavelet Packets and HVS Modeling.** The research done in this thesis made a particularly important contribution in investigating the effects of combining wavelet packet decomposition with HVS characteristic modeling. These two areas are often treated as separate techniques. Incorporating them into one codec has important implications for quality assessment due to the interactions between the wavelet transform and HVS modeling. Among these implications is the fact that measured PSNR can actually *decrease* with *increased* perceptual quality.

5. **Quality Assessment.** We have examined the area of quality assessment of the reconstructed image, particularly the application of HVS modeling in assessment tools. We have stressed throughout that it is important to apply both *quantitative* and *qualitative* assessment mechanisms. Two frequency-weighted peak signal-to-noise ratio (WPSNR) quality measures were introduced. The squared-error distortion maps derived from quality measures were also used in presenting and comparing compression performance.

6. **Thorough Experiments.** We have investigated and reported the compression results on a large variety of images of different types. Our experimental image library contained 54 images, which is much larger than most reported experiments. This variety of images was used to examine the performance differences from varying the WP/HVS components, including an unusual direct comparison of uniform and non-uniform quantization within the same codec. We used a unique "greedy" design methodology to stream-line the WP/HVS framework into the "best" possible codec. The performance was compared to other codecs using both quantitative *and* qualitative analysis. We categorized the test images in order to form constructive generalizations of framework results.

7. **A Quality Codec.** The final product of this research work includes a practical, general-purpose image codec, which incorporates a model of the HVS. The development of an acceptable distortion factor (ADF) system facilitates a user-friendly product. Many systems require the user to set a desired compression bit-rate, which is not convenient since, as we have seen, compressing different images at the same bit-rate can yield very different qualities. The codec, by default, maximally compresses an image without introducing perceptual distortion. This convenience is a strong benefit of the WP/HVS codec. Higher compression rates can easily be achieved by specifying a higher ADF, in which case distortion

will be added more quickly into higher-frequency, diagonally-oriented signals, which is less noticeable to the HVS than distributing the error evenly over all frequencies. The ADF technique is also far superior to using a global threshold based only on wavelet coefficient size.

The flexible, computationally-light, wavelet packet decomposition scheme allows greater compression on certain types of images with unusual power spectra, and also performs well on more typical images. Wavelet packets are adaptive to image signal content and yet still can support the standard pyramidal wavelet decomposition scheme.

6.4 Thesis Limitations

There are a few limitations of the research and development approach taken in this thesis which need to be specified.

1. **Codec Design Methodology.** Our greedy codec design approach was one practical technique that could be used to proceed from the WP/HVS framework to a WP/HVS codec. It allowed independent experimentation with framework components. Without independently testing components, there would be far too many parameters to examine at once. However, this approach ignores possible interactions between components and therefore cannot guarantee an optimal codec. Note that we did examine the interactions between the non-uniform quantizer and the symbol encoder.

2. **Result Comparison Technique.** We developed a simple quality over rate (Q/R) ratio which could be averaged over image category to compare results with different framework configurations on a level playing field. The results from such a comparison were used to make decisions about which WP/HVS components should be kept. The Q/R ratios could only be considered reliable for comparing techniques at *similar* bit-rates due to the nature of the changing slopes of rate-distortion curves. Even under this restriction, "close" Q/R ratios should not be assumed to represent a significant performance difference between techniques.

3. **Masking.** We have not been able to explicitly model the HVS masking effect. The non-uniform quantizer was designed to do so, but its results were not encouraging. However, when using qualitative assessment tools to examine

the codec results, we found indications that the masking effect was being implicitly modeled by interactions between the CSF filtering pre-processing and the uniform quantizer designed with HVS sensitivity in mind.

4. **Quality Assessment.** The weighted PSNR quality measures presented in this thesis are a definite improvement over PSNR, since they do attempt to incorporate aspects of the HVS. More comprehensive measures could be and have been developed, but often these require drastic increases in computational complexity without complementary improvements in correlation with the HVS.

6.5 Suggestions for Future Research

The image compression research and results presented in this thesis are certainly limited, and represent only a small fraction of possible work in the area of wavelet packets and human visual system characteristics. There are many areas in particular which deserve further attention.

1. **CSF Filtering.** The WP/HVS codec applies CSF filtering as a fast Fourier transform pre-processing step. Filtering in this manner is convenient for research experiments and also due to the fact that the CSF itself was developed from experiments in the Fourier domain. We have found that approximately 75% of compression time is taken by the CSF filtering stage. The ability to incorporate the CSF into the wavelet packet transform itself should reduce computational requirements significantly and be a more elegant solution to modeling HVS spatial frequency sensitivity.

2. **Zero Trees.** One of the most powerful ideas used in many wavelet-based compression systems is the zero tree [53, 51]. A tree or sub-tree in the wavelet sub-band decomposition consisting entirely of zeros or insignificant coefficients can be represented extremely efficiently by one symbol using this technique. Incorporating zero trees into the WP/HVS codec could increase performance substantially, without a significant increase in computational load.

3. **Modeling Masking.** Our attempt to model masking with a non-uniform quantizer was not particularly effective, probably due to interactions with the arithmetic coder. It is possible that visual masking could be modeled more successfully with a method other than non-uniform scalar quantization. Perhaps the use of vector quantization should be explored in this realm, since dealing with

vectors or blocks of wavelet coefficients may be a more reliable way to detect how the presence of a certain signal may mask the visibility of distortion.

4. **Human Observer Experiments.** To better judge the qualitative results of the WP/HVS codec, experiments involving human observers could be conducted. These experiments may involve groups of people, a large variety of images (e.g., 50), and different codecs. The mean opinion score (MOS) could be found to judge the various codecs [42]. The results from such an experiment may indicate which portions of the WP/HVS codec require more work and also could be used to test the efficacy of the weighted PSNR measures.

5. **Progressive Transmission.** We have not examined the applicability of the WP/HVS codec to progressive transmission. As briefly discussed in Chapter 2, wavelet-based techniques are well-suited to progressive transmission due to their multiresolution sub-band nature. It is likely that incorporating wavelet packets and HVS characteristics may improve progressive transmission, allowing a higher quality image to be retrieved more quickly.

6. **Other Test Images.** Fingerprint and Structured Texture images were found to benefit from the use of a flexible wavelet packet decomposition as opposed to the pyramidal wavelet transform. It may be worthwhile to examine other special types of images to which wavelet packets would be successfully applied.

7. **Encryption.** One final item for future research involves the incorporation of encryption technology into the WP/HVS codec. Some situations may require the secure transmission of images over an otherwise insecure network.

Bibliography

[1] N. Akrout, R. Prost, and R. Goutte. Image compression by vector quantization: A review focused on codebook generation. *Image and Vision Computing*, 12(10):627-637, 1994.

[2] M. G. Albanesi. Wavelets and human visual perception in image compression. In *International Conference on Pattern Recognition*, volume 13, pages 850-863, Technical University of Vienna, Austria, 1996.

[3] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies. Image coding using wavelet transform. *IEEE Transactions on Image Processing*, 1(2):205-220, April 1992.

[4] A. P. Bradley, April 1997. Personal communication.

[5] C. M. Brislawn. Fingerprints go digital. *Notices of the American Mathematical Society*, 42(11):1278-1283, November 1995.

[6] C. M. Brislawn, J. N. Bradley, R. J. Onyshczak, and T. Hopper. The FBI compression standard for digitized fingerprint images. In *SPIE Proceedings*, volume 2847, August 1996.

[7] P. Brodatz. *Textures: A Photographic Album for Artists and Designers.* Dover Publications, Incorporated, New York, 1966.

[8] S. E. Budge, T. G. Stockham Jr., D. M. Chabries, and R. W. Christiansen. Vector quantization of color digital images within a human visual model. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 816-819, April 1988.

[9] C.-K. Chan and C.-K. Ma. A fast method of designing better codebooks for image vector quantization. *IEEE Transactions on Communications*, 42(2/3/4):237-242, February 1994.

[10] T. Chang and C.-C. J. Kuo. Texture analysis and classification with tree-structured wavelet transform. *IEEE Transactions on Image Processing*, 2(4):429-441, October 1993.

[11] H. Cheng and X. Li. Image Compression and Partial Encryption Based on Tree Structures. *IEEE Transactions on Communications*, submitted 1996.

[12] C. Chinnock. Wavelets challenge MPEG, fractals. *BYTE*, page 34, December 1995.

[13] R. J. Clarke. *Digital Compression of Still Images and Video.* Academic Press, San Diego, California, 92101, 1995.

[14] Y. Cohen, M. S. Landy, and M. Pavel. Hierarchical coding of binary images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(3):284–298, May 1985.

[15] R. R. Coifman and M. V. Wickerhauser. Entropy-based algorithms for best basis selection. *IEEE Transactions on Information Theory*, 38(2):713–718, March 1992.

[16] P. J. Cordell and R. J. Clarke. Low bit-rate image sequence coding using spatial decomposition. *IEEE Proceedings-I, Communications, Speech, and Vision*, 139(6):575–581, December 1992.

[17] D. Cortez, P. Nunes, M. M. de Sequiera, and F. Pereira. Image segmentation towards new image representation methods. *Signal Processing. Image communication*, 6(6):485–498, 1995.

[18] P. C. Cosman, K. L. Oehler, E. A. Riskin, and R. M. Gray. Using vector quantization for image processing. *Proceedings of the IEEE*, 81(9):1326–1341, September 1993.

[19] S. Daly. The visible difference predictor: An algorithm for the assessment of image fidelity. In A. B. Watson, editor, *Digital Images and Human Vision*, chapter 14, pages 179–206. Bradford Book, Massachusets Institute of Technology, 1993.

[20] E. X. DeJesus. Walking, talking web. *BYTE*, 22(1):80NA1–80NA4, January 1997.

[21] M. R. El-Sakka and M. S. Kamel. A segmentation criterion for digital image compression. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 2551–2554, 1995.

[22] W. H. Equitz. A new vector quantization clustering algorithm. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(10):1568–1575, October 1989.

[23] A. M. Eskicioglu and P. S. Fisher. A survey of quality measures for gray scale image compression. In *NASA Space Earth Science Data Compression Workshop*, pages 49–61, 1993.

[24] S. Fioravanti F. De Natale, G. S. Desoli and D. Giusto. An edge-based splitting criterion for adaptive transform coding. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 409–412, 1993.

[25] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practice.* The Systems Programming Series. Addison-Wesley Publishing Company, second edition, 1992.

[26] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression.* Kluwer Academic, The Netherlands, 1992.

[27] A. Gersho and M. Yano. Adaptive vector quantization by progressive code vector replacement. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 133–136, 1985.

[28] R. C. Gonzalez and R. E. Woods. *Digital Image Processing.* Addison-Wesley Publishing Company, 1993.

[29] R. M. Gray. Vector quantization. *IEEE Acoustics, Speech and Signal Processing Magazine*, 1(2):4–29, 1984.

[30] C.-M. Huang and R. W. Harris. A comparison of several vector quantization codebook generation approaches. *IEEE Transactions on Image Processing*, 2(1):108–112, January 1993.

[31] Y. Kim, I. Choi, I. Lee, T. Yun, and K.-T. Park. Wavelet transform image compression using human visual characteristics and a tree structure with a height attribute. *Optical Engineering. Society of Photo-optical Instrumentation Engineers*, 35(1):204–212, January 1996.

[32] S. A. Klein. Image quality and image compression: A psychophysicist's viewpoint. In A. B. Watson, editor, *Digital Images and Human Vision*, chapter 7, pages 73–88. Bradford Book, Massachusets Institute of Technology, 1993.

[33] J. Knipe, X. Li, and B. Han. An improved lattice vector quantization based scheme for wavelet compression. *IEEE Transactions on Signal Processing*, submitted September 1996.

[34] M. Kunt, A. Ikonomopolous, and M. Kocher. Second generation image coding techniques. *Proceedings of the IEEE*, 73(4):549–574, April 1985.

[35] O.-J. Kwon and R. Chellappa. Segmentation-based image compression. *Optical Engineering. Society of Photo-optical Instrumentation Engineers*, 32(7):1581–1587, July 1993.

[36] A. S. Lewis and G. Knowles. Image compression using the 2-D wavelet transform. *IEEE Transactions on Image Processing*, 1(2):244–250, April 1992.

[37] Y. Linde, A. Buzo, and R. M. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, 28(1):84–95, 1980.

[38] T. D. Lookabaugh and R. M. Gray. High-resolution quantization theory and the vector quantizer advantage. *IEEE Transactions on Information Theory*, 35(5):1020–1033, September 1989.

[39] J. Lu, V. R. Algazi, and R. R. Estes Jr. Evaluation and synthesis of wavelet image coders. In *International Conference on Image Processing*, volume 1, pages 590–593, 1995.

[40] J. L. Mannos and D. J. Sakrison. The effects of a visual fidelity criterion on the encoding of images. *IEEE Transactions on Information Theory*, 20(4):525–536, July 1974.

[41] S. Marshall. Application of image contours to three aspects of image processing: compression, shape recognition and stereopsis. *IEEE Proceedings-I, Communications, Speech, and Vision*, 139(1):1–8, February 1992.

[42] M. Miyahara, K. Kotani, and V. R. Algazi. Objective picture quality scale (PQS) for image coding. Technical report, Center for Image Processing and Integrated Computing, 1996. Submitted to *IEEE Transactions on Communications.*

[43] S. A. Mohamed and M. M. Fahmy. Image compression using block pattern-vector quantization. *Signal Processing: EURASIP*, 34(1):69–84, October 1993.

[44] N. M. Nasrabadi and R. A. King. Image coding using vector quantization: A review. *IEEE Transactions on Communications*, 36(8):957–971, August 1988.

[45] Graphics Technology Department of CompuServe Incorporated. Graphics Interchange Format Version 89a. Technical report, CompuServe Incorporated, July 1990.

[46] T. P. O'Rourke and R. L. Stevenson. Human visual system-based wavelet decomposition for image compression. *Journal of visual communication and image representation*, 6(2):109–121, June 1995.

[47] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press, second edition, 1995.

[48] K. Ramchandran and M. Vetterli. Best wavelet packet bases in a rate-distortion sense. *IEEE Transactions on Image Processing*, 2(2):160–175, April 1993.

[49] A. Rosenfeld and A. C. Kak. *Digital Picture Processing.* Academic Press, 1976.

[50] T. W. Ryan, L. D. Sanders, H. D. Fisher, and A. E. Iverson. Image compression by texture modeling in the wavelet domain. *IEEE Transactions on Image Processing*, 5(1):26–36, January 1996.

[51] A. Said and W. A. Pearlman. A new fast and efficient image codec based on set partitioning in hierarchical trees. *IEEE Transactions on Circuits and Systems for Video Technology*, 6(3):243–250, June 1996.

[52] D. Sampson and M. Ghanbari. Fast lattice-based gain-shape vector quantisation for image-sequence coding. *IEE Proceedings-I, Communications, Speech, and Vision*, 140(1):56–66, February 1993.

[53] J. M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Transactions on Signal Processing*, 41(12):3445–3462, December 1993.

[54] H. L. Snyder. The visual system: Capabilities and limitations. In L. E. Tannas Jr., editor, *Flat-Panel Displays and CRTs*, chapter 3, pages 54–69. Van Nostrand Reinhold, New York, 1985.

[55] L. Torres, J. R. Casas, and S. de Diego. Segmentation based coding of textures using stochastic vector quantization. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 597–600, 1994.

[56] J. Vaisey and A. Gersho. Image compression with variable block size segmentation. *IEEE Transactions on Signal Processing*, 40(8):2040–2060, August 1992.

[57] G. K. Wallace. The JPEG still picture compression standard. *IEEE Transactions on Consumer Electronics*, 38(1):xviii, February 1992.

[58] A. B. Watson, G. Y. Yang, J. A. Solomon, and John Villasenor. Visual thresholds for wavelet quantization error. In B. Rogowitz and J. Allebach, editors, *SPIE Proceedings*, volume 2657, pages 382–392, 1996.

[59] A. B. Watson, G. Y. Yang, J. A. Solomon, and John Villasenor. Visibility of wavelet quantization noise. *IEEE Transactions on Image Processing*, to appear.

[60] S. J. P. Westen, R. L. Lagendijk, and J. Biemond. Perceptual image quality based on a multiple channel HVS model. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 2351–2354, 1995.

[61] R. Wilson. Quad-tree predictive coding: a new class of image data compression algorithms. *Proceedings of the 1984 IEEE Conf. on ASSP*, pages 29.3.1–29.3.4, 1984.

[62] X. Wu. Image coding by adaptive tree-structured segmentation. *IEEE Transactions on Information Theory*, 38(6):1755–1767, November 1992.

[63] X. Wu and Y. Fang. A segmentation-based predictive multiresolution image coder. *IEEE Transactions on Image Processing*, 4(1):34–47, January 1995.

[64] Z. Xiong, K. Ramchandran, and M. T. Orchard. Space-frequency quantization for wavelet image coding. *IEEE Transactions on Image Processing*, 6(5), May 1997.

[65] Z. Xiong, K. Ramchandran, and M. T. Orchard. Wavelet packets image coding using space-frequency quantization. *IEEE Transactions on Image Processing*, submitted 1996.

[66] W. Xu and G. Hauske. Perceptually relevant error classification in the context of picture coding. In *Image Processing and its Applications*, pages 589–593, July 1995.

[67] P. Zhou, M. Khansari, and A. Leon-Garcia. Noncausal predictive image coding. In *SPIE Proceedings*, volume 2094, pages 789–800, 1993.

Appendix A

WP/HVS Framework Implementation Summary

A.1 Introduction

This appendix contains a brief description of the WP/HVS compression framework implementation, including the source code, C++ classes, and executable programs.

A.2 Source Code

The framework has been implemented using C++ on the UNIX operating system. It is based on the "Baseline Wavelet Transform Coder Construction Kit" written by Geoff Davis (gdavis@cs.dartmouth.edu), John Danskin, and Ray Heasman. The code for the construction kit can be found at

http://www.cs.dartmouth.edu/~gdavis/wavelet/wavelet.html.

Additionally, the weighted PSNR measures are based on the Picture Quality Scale code, entitled "CIPIC PQS version 1," which can be found in the

ftp://info.cipic.ucdavis.edu/pub/cipic/code/pqs/

directory. It was written by Robert Estes (estes@cipic.ucdavis.edu) and V. Ralph Algazi (vralgazi@ucdavis.edu).

The above code was modified and extended (by about 3 000 lines) to implement wavelet packets, non-uniform quantization, and CSF filtering, bringing the total close to 10 000 lines of documented C++. The documentation is also available in HTML format, thanks to the cxx2html program written by Darrell Schiebel (drs@nrao.edu), available from

ftp://aips2.cv.nrao.edu/pub/aips++/RELEASED/cxx2html/.

A.3 C++ Classes

The most important classes relevant to the framework are shown in Figure A.1.[1] The following summarizes the functionality of the depicted classes.

Image. Handles manipulation of grayscale images and representations of wavelet coefficients. Support for file I/O, CSF and weighted PSNR filtering operations, and calculation of wavelet packet decomposition stopping criteria. The file type used by default is PGM (portable graymap) developed by Jef Poskanzer.

Quantizer. Abstract class, top of the quantizer hierarchy.

UniformQuant. A single-layer, non-embedded quantizer. Maps each coefficient to a single symbol which is later entropy coded. The fineness of the quantization can be controlled.

MaskingQuant. Handles incorporation of HVS sensitivity, including the masking effect, into the quantization process. The MaskingQuant class inherits from the UniformQuant class, but is extended to set the stop-band and the acceptable distortion factor (ADF). Another important purpose of this class is to allow experimentation with using a non-uniform quantizer to take advantage of the masking effect.

Matrix. This class implements a simple matrix data structure and includes some operations useful to matrices, including least-squares fitting of polynomials. Within the WP/HVS framework, this class is only relevant for attempting to fit a function to model the non-linear quantizer bin-size parameters.

WaveletTransform. High-level wavelet transform implementation. Breaks wavelet transformed images up into sub-bands. This makes post-processing more convenient and also independent of the method of transform.

PacketTransform. Analogous to the WaveletTransform class, but using wavelet packet decomposition.

[1] The Object class is included only for conceptual reasons and could be implemented as an abstract base class.

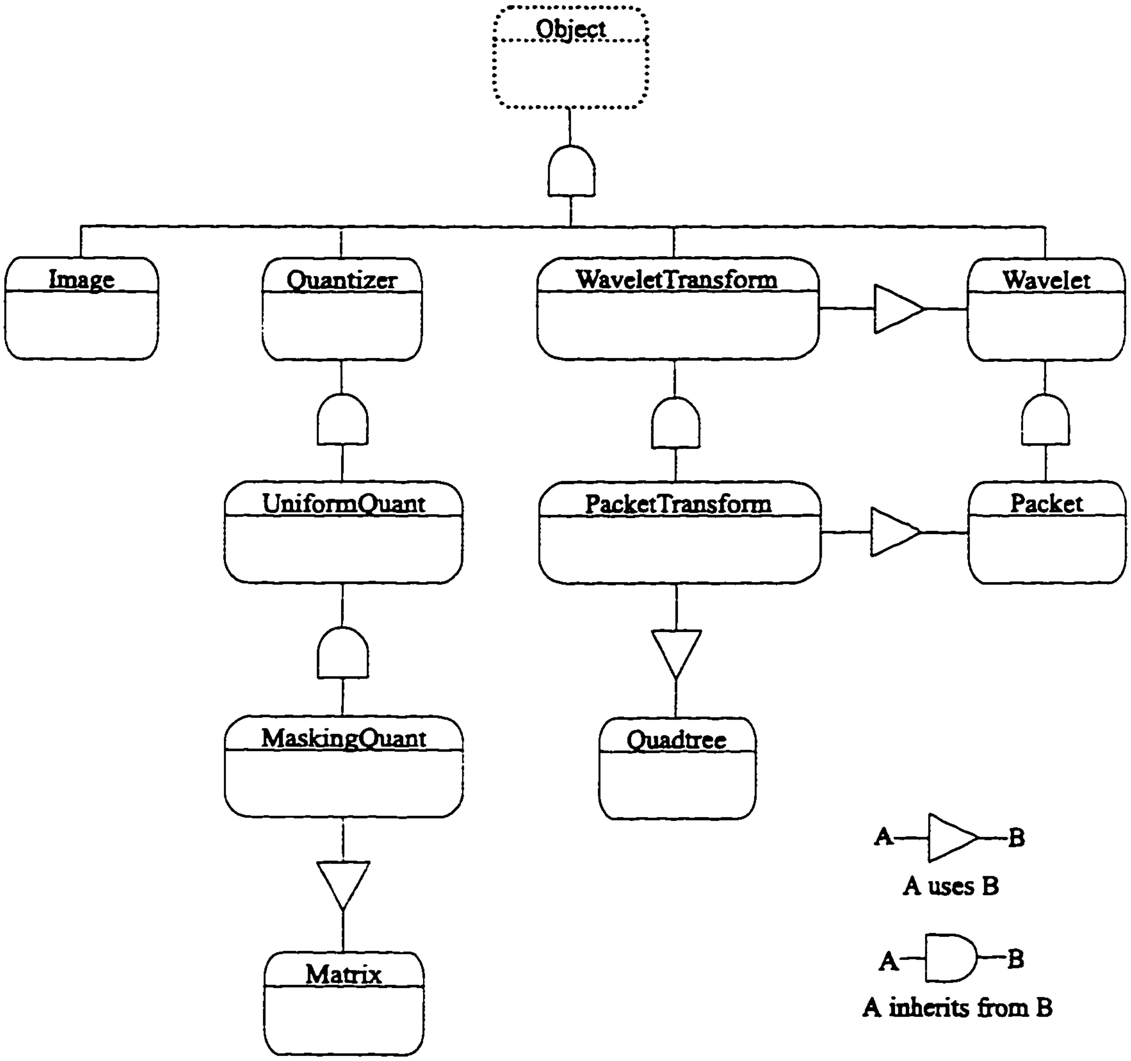


Figure A.1: Subset class hierarchy of the WP/HVS framework.

Quadtree. Handles manipulating the quadtree structure representation of wavelet packet decomposition.

Wavelet. Low level wavelet transform routines; performs the actual transform of the image into coefficients of basis functions.

Packet. Analogous to the **Wavelet** class, but using wavelet packet decomposition.

A.4 Executables

There are three executable programs associated with the WP/HVS framework. All three have thus far been implemented as text-based, command-line programs.

Encode. Takes an input image in raw PGM format and compresses it using the framework.

Decode. Takes an input file of compressed data and decompresses it, yielding the reconstructed image in PGM format.

Quality. Implements the quality assessment stage. Given the original and reconstructed images (both in PGM format), **Quality** outputs the PSNR, the two weighted PSNRs, and the scaled error images from all three quality measures.