Lossy Compression of Medical Ultrasound Images Using Space-Frequency Segmentation

by

Edwin Chiu

B.A.Sc. University of British Columbia, 1988

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE
in the School
of
Engineering Science

SIMON FRASER UNIVERSITY

July 1999

Approval

Name: Edwin Chiu

Degree: Master of Applied Science

Title of thesis: Lossy Compression of Medical Ultrasound Images Using Space-Frequency Segmentation

Examining Committee:

Dr. Rick Hobson
Professor of Engineering Science, SFU
Chairperson

Dr. Jacques Vaisey
Associate Professor of Engineering Science, SFU
Senior Supervisor

Dr. M. Stella Atkins
Professor of Computing Science, SFU
Supervisor

Dr. Ze-Nian Li
Professor of Computing Science, SFU
Examiner

Date Approved:

Abstract

Ultrasound is a useful and popular medical imaging modality. Due to the large volume of images generated, storage and communications issues motivate research into ultrasound image compression. Lossy techniques achieve higher compression than lossless methods, but must preserve diagnostically relevant information and maintain acceptable aesthetic quality.

This thesis applies the technique of space-frequency segmentation using balanced wavelet packet trees to the lossy compression of medical ultrasound images. In order to represent an image, space-frequency segmentation searches a large family of possible space-frequency partitions, together with sets of possible quantizers, and finds the rate-distortion optimal combination. Zerotree coding is a leading high performance lossy compression technique, but because it uses a fixed frequency decomposition topology designed for natural images and allocates bits based only on coefficient magnitude, it produces distinctive artifacts at higher compression rates when applied to ultrasound images.

Space-frequency segmentation has not been previously applied to medical ultrasound images, and unlike other published results, real entropy coders are implemented to code the space-frequency representation. Space-frequency segmentation improves both objective and subjective quality when compared against zerotree coding using Set Partitioning In Hierarchical Trees (SPIHT), and since the technique is not yet mature, there remain many possibilities to further enhance the performance. The improvement in subjective quality using space-frequency segmentation is confirmed by a study in which ultrasound radiologists assess the quality of compressed images with respect to diagnostically

relevant features. The optimum space-frequency partition highlights the characteristic speckle texture of ultrasound images as a particularly difficult feature to compress, and leads to a more effective decomposition topology for preserving speckle.

Acknowledgments

The author would like to thank Dr. Anne R. (Jean) Buckley of the Department of Radiology at Vancouver General Hospital for her help in conducting the expert viewer assessment study. This research was partially funded by the Natural Sciences and Engineering Research Council of Canada, the BC Advanced Systems Institute, and Simon Fraser University.

Contents

List of Tables

List of Figures

List of Abbreviations

bpp	bits per pixel
BFOS	Breiman, Friedman, Olshen and Stone
codec	coder and decoder
CQF	conjugate quadrature filter
CT	computed tomography
CTWS	continuous time wavelet series
CWT	continuous wavelet transform
DCT	discrete cosine transform
EZW	embedded zerotree wavelet
FIR	finite impulse response
JPEG	Joint Photographic Experts Group
KLT	Karhunen-Loève transform
MRI	magnetic resonance imaging
MSB	most significant bit
MSE	mean squared error
PACS	picture archiving and communications systems

PCM	pulse coded modulation
pdf	probability density function
PSNR	peak signal to noise ratio
PTSVQ	pruned tree structured vector quantization
QMF	quadrature mirror filter
ROC	receiver operating characteristic
SPIHT	set partitioning in hierarchical trees
TGC	time gain compensation
VGH	Vancouver General Hospital

Chapter 1

Introduction

Diagnostic medical ultrasonography refers to the use of echoes from ultrasonic mechanical waves to view structures within the body. Ultrasound is a popular medical imaging modality because it is non-invasive, versatile, there are no known side effects, and the equipment used for ultrasonic scanning is small and inexpensive relative to other options. Many people are familiar with ultrasound from its applications in obstetrics, where it is used to examine the fetus at different stages during pregnancy. Ultrasound is also commonly used to examine the liver, kidneys, and other abdominal organs. Another field of ultrasonography is echocardiography.

With the advent of real-time ultrasound transducers, ultrasound exams produce video output; however, still images captured by the sonographer during an exam are still widely used for diagnosis, and generally serve as the archival record of the examination. Because of the way ultrasound images are acquired, they are stored in digital format. A typical exam can produce twenty or thirty images, which means that a large volume of digital image data is generated by a diagnostic ultrasound installation. Reducing storage requirements and making access to data more convenient are two of the motivations for applying compression to ultrasound images. Storage and access issues have become more important with the recent development of medical picture archiving and communications systems (PACS), which attempt to provide an integrated facility for storage, communication and display of the images from different modalities at a single site. Another new technology driving interest in medical image compression is telemedicine, where remote consultation requires the transmission of images over limited bandwidth channels.

Lossless methods do not provide enough compression for medical images. Lossy methods provide higher compression, but result in loss of information. Retaining diagnostically relevant information is the most important goal for a lossy medical image compression scheme; however, producing an aesthetically acceptable image is also a requirement. There are many legal and standardization issues that still need to be addressed before lossy compression can be widely deployed in medical applications. Research into compression methods drives the acceptance of lossy compression for medical images before standards are established, and leads to the implementation of practical systems after standards are established.

Subband coding of images is a technique that separates an image into different frequency bands for effective coding. Separation into different bands reduces redundant information, and allows bands to be coded differently depending on their statistics. A fixed filter topology is desirable because of its simplicity, but the fixed topology chosen should suit the characteristics of the image to be coded. In a subband coding context, the wavelet transform commonly (though imprecisely) refers to a fixed filter topology in which the filter width decreases logarithmically by a factor of two with decreasing frequency. The wavelet transform concentrates energy in the low pass bands, and is effective for coding natural images, which generally have little high frequency content.

Ultrasound images have some unique characteristics that distinguish them from natural images. An unusual feature of ultrasound images is the speckle texture that is caused by the underlying physics of ultrasonic imaging. Compared to the energy distribution of natural images, speckle results in more energy in certain high pass subbands. Another characteristic of ultrasound images is the spatial variation in statistics across an individual image. An image consists of an ultrasound scanned area, which is often non-rectangular, against a passive background, which may contain text and limited graphics. Because ultrasound images differ from natural images, subband coding techniques that use fixed filter topologies designed for natural images may not be the best choice for ultrasound images. An alternative is to use an image adaptive scheme.

When it is difficult to make assumptions about the image to be compressed, a fixed filter topology may not be suitable. One solution is an adaptive filtering scheme that selects the best filter topology for each image. Viewing the decomposition of an image into subbands as the projection onto a choice of basis functions, the problem is to choose an appropriate basis to represent an individual image. The drawbacks to an adaptive scheme include the computational cost of finding the best basis, and the need to send a description of the basis as side information. The algorithm for finding the optimum representation must have a meaningful criterion for choosing the best basis, and must limit the choice of basis to a useful but finite set of possibilities.

Space-frequency segmentation finds the optimum basis to represent an image from a large family of possible bases. As the name suggests, space-frequency segmentation involves the decomposition of an image using a hierarchy of space and frequency partitions. Space partition refers to the division of an image into four spatial quadrants, and frequency partition refers to the division of an image into four frequency subbands. The subimages produced by space or frequency partitioning can also be successively partitioned in space or frequency, so that a decomposition tree of space and frequency partitions is built, up to some maximum depth. The algorithm is symmetric in terms of space and frequency partitions – decomposition in either space or frequency is allowed at each branch of the decomposition hierarchy. Allowing both space or frequency partitions on any subimage permits more basis choices than previous related methods, such as single-tree wavelet packets, which is limited to the same filter topology over the entire image, and double-tree wavelet packets, which is limited to different filter topologies over spatial regions of the original image only. Quantizers for a subimage are chosen from a finite set. Space-frequency segmentation chooses the optimal combination of space and frequency partitions, and the best quantizer for each subimage, according to a rate-distortion criterion. For a fixed target rate, the algorithm finds the combination of partitions and quantizers that produces the minimum distortion representation. The optimization incorporates a fast tree pruning algorithm for efficient computation.

This thesis studies the compression of grayscale medical ultrasound images using space-frequency segmentation. In the implemented space-frequency codec, uniform scalar quantization of subimages is followed by entropy coding. Different entropy coding methods for space-frequency segmented images are investigated, and real entropy coders are implemented to code ultrasound images. The tradeoffs between different codec parameters are studied to find a space-frequency segmentation codec that works well with ultrasound images. Because there are no widely accepted medical ultrasound test images, the performance of space-frequency segmentation is compared against a current leading wavelet transform coding algorithm, Set Partitioning In Hierarchical Trees (SPIHT). In addition to the use of a squared-error metric to quantify objective quality, subjective quality is used to judge compressed images. Medical ultrasound images are difficult for non-expert viewers to interpret. A study is conducted in which ultrasound radiologists assess the subjective quality of compressed images with respect to medically relevant features. The expert viewer study shows the improved quality of space-frequency compressed images compared to SPIHT compressed images. Investigation of the optimum space-frequency partition leads to a fixed partition that is better suited to the speckle texture of ultrasound images.

Space-frequency segmentation has not been used previously to compress medical ultrasound images. The goal of the this thesis is to apply space-frequency segmentation to ultrasound images, to understand the issues involved, and to assess the performance of this technique on these images. The experiments conducted are designed to give insight into the problem, and to give understanding of the tradeoffs required for an effective space-frequency codec. The investigations are not exhaustive, but are complete enough to show that space-frequency segmentation is a promising method for compressing medical ultrasound images.

1.1 Thesis outline

Chapter 2 presents an overview of ultrasound image generation, discussing ultrasound physics and transducers.

Chapter 3 starts by discussing source coding techniques for image compression, reviewing transform coding and subband coding. Next, the connection between subband coding and wavelets is explained. Finally, image coding schemes resulting from the recent research in wavelets and subband coding are presented, including zerotree coding and wavelet packets.

Chapter 4 connects image coding with ultrasound images by reviewing some recent articles on lossy ultrasound image compression, and presenting a simple experiment comparing a natural image and an ultrasound image after subband decomposition.

Chapter 5 explains the theory behind space-frequency segmentation. Because it is a key component of the space-frequency segmentation algorithm, the rate-distortion optimization technique of Ramchandran and Vetterli is discussed. Space-frequency segmentation is examined in detail. The basis choices and basis selection algorithm are explained.

Chapter 6 reports the experimental results. Coding schemes for use with space-frequency segmentation for normal and ultrasound images are investigated. Space-frequency segmentation of ultrasound images is evaluated using objective and subjective measures, and ultrasound images compressed using space-frequency segmentation are assessed by expert viewers. The space-frequency representation of ultrasound images is studied.

Chapter 7 presents the conclusions and suggests future work.

Chapter 2

Diagnostic Ultrasound

2.1 Introduction

Diagnostic ultrasound refers to the use of echoes from pressure waves to obtain information about tissue for diagnostic purposes. By piezoelectric effect, an electrical signal is converted to a mechanical signal at a transducer, and the mechanical signal is applied to a tissue medium. This ultrasound signal is generally pulsed. As the pulse propagates through the tissue, structures in the tissue produce reflections that travel back to the transducer. At the transducer, these mechanical echoes are converted back to electrical signals. The strength of a reflected signal contains information about the reflecting structure, and the delay between sending a signal and receiving an echo indicates the distance between the structure and the transducer.

The simplest way to present the results of scanning in a single direction is to display the amplitude of the echo plotted against time, like a signal trace on an oscilloscope. This is known as A-mode, and clearly has limited utility. By taking multiple scans at regular intervals and displaying the strength of an echo at a location as the brightness of a spot, B-mode scanning produces a two-dimensional grayscale image. A typical sector scanned B-scan image of a kidney[1] is shown in Figure 2.1. The scan rates of modern transducers

1. This image is obtained from the ALI Technologies website, www.alitech.com. ALI is a manufacturer of medical PACS.

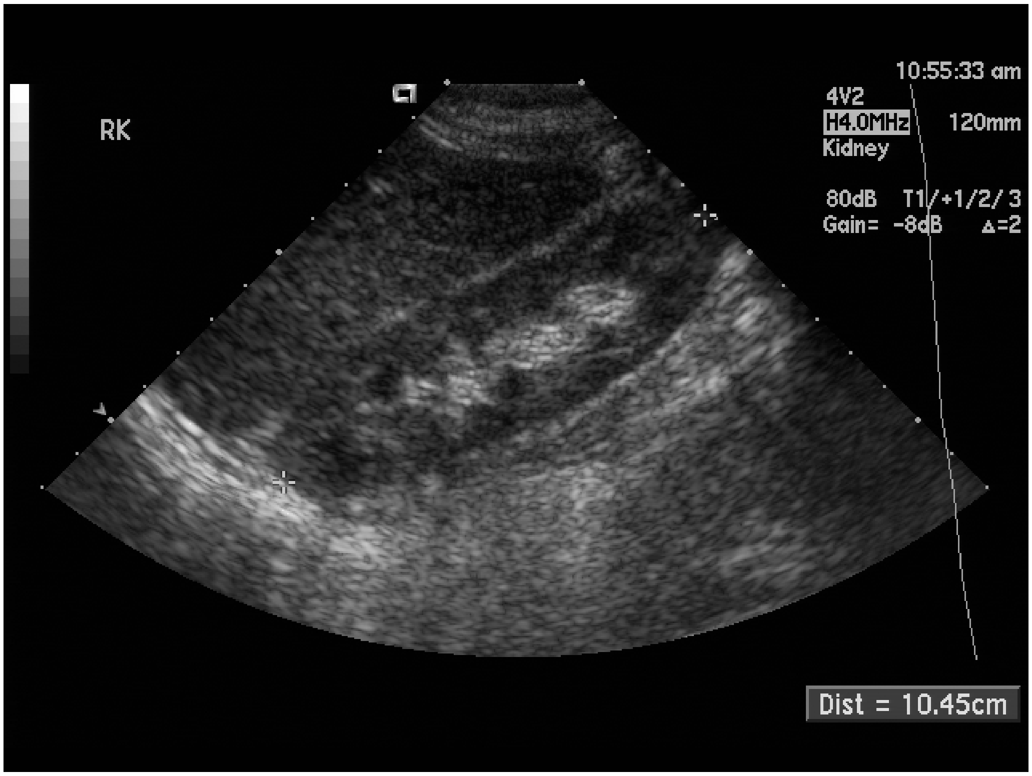


Figure 2.1: A typical ultrasound image

allow real time images to be generated, displaying motion over time as well as structure. Not all ultrasound images are completely grayscale. For example, the Doppler shift of the received echoes can be used to measure the flow of a moving substance such as blood, and the velocity of the flow can be displayed as a color overlay. Nevertheless, pure grayscale images captured at a single instant in time are very common and are useful for diagnostic purposes. The scope of this thesis is limited to the study of lossy compression of grayscale ultrasound B-scan images.

Ultrasound images exhibit a characteristic texture called speckle. Depending on the context and application, speckle in medical images used for diagnostic purposes can be viewed as signal or noise. For example, speckle can be used to characterize tissue [29] [56], or speckle can mask diagnostically relevant features [43] [60]. Good performance in a variety of contexts is desirable for a compression system targeted for medical ultrasound images. When it is not known a priori whether speckle is viewed as signal or noise in a specific application, the compression algorithm needs to preserve it. The presence of speckle in an image is also something which radiologists are accustomed to. The compression scheme should avoid altering the image in a noticeable way, so for aesthetic reasons speckle should be preserved even when it does not provide direct diagnostic information. The presence of speckle and the desire to preserve it are issues that distinguish the ultrasound image compression problem from the compression of natural images.

Ultrasound images look quite different from natural images. These differences result from the way ultrasound images are generated – echo-location using mechanical waves at much longer than optical wavelengths results in images that appear quite alien to non-expert observers. Some background material on how ultrasound images are generated is useful before tackling the problem of how to compress these images. The cursory discussion of ultrasound physics and transducers that follows, is drawn from two books by Fleischer and James Jr. [21] [22], and a book by McDicken [34]. This background material explains the appearance of ultrasound images, and the representation of medically relevant features in the images.

2.2 Ultrasound physics

The speed of propagation of ultrasound, s, is defined in terms of the frequency f and wavelength λ by

$$s = f\lambda . \tag{2.1}$$

The speed of propagation varies depending on the density ρ and compressibility of the medium; in air it is 330 m/s, whereas in bone the speed is 3500 m/s. In most tissue, s is approximately 1540 m/s, and this value is typically used by the scanning equipment to convert time to distance.

The incident pulse is reflected when it encounters an interface between two materials with different density. The reflectivity coefficient R relates the amplitude of the reflected signal to the amplitude of the incident signal,

$$R = \frac{Z_1 - Z_2}{Z_1 + Z_2}, \tag{2.2}$$

where Z is the acoustical impedance,

$$Z = \rho s. \tag{2.3}$$

An interface where there is a significant impedance discontinuity produces a strong reflection, which appears bright in the image. Ultrasound radiologists are trained to identify a physiological feature using the intensity with which it is represented along with the context in which it appears. Specular reflection describes reflection from an interface that is large with respect to the wavelength, and it is this type of reflection that produces a feature in a B-scan image. If the texture of the reflecting interface is small with respect to the wavelength, the result is scattered reflection. The strength of scattered reflection received at the transducer is much lower that of specular reflection. Scattered reflection produces a distinctive grainy texture in the image known as speckle.

If the reflecting interface is not perpendicular to the incident signal, the strength of the echoes received at the transducer is reduced. Deviations of as little as 3° from perpendicular incidence produce significant reduction in echo amplitude. The incident signal is also refracted as it passes from one material to another according to Snell's Law,

$$\frac{s_1}{s_2} = \frac{\sin\theta_i}{\sin\theta_r}, \tag{2.4}$$

where θ_i is the angle of incidence and θ_r is the angle of refraction. Refraction is illustrated in Figure 2.2. Refraction requires the sonographer to adjust the orientation of the transducer to obtain the best view of an organ.

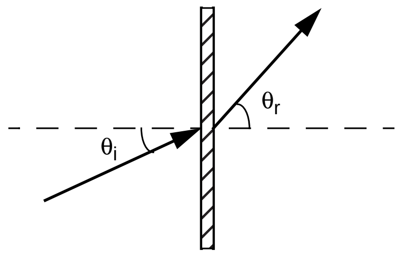


Figure 2.2: Refraction of the incident signal

The signal is also attenuated as it propagates through the insonified object. This attenuation is caused by absorption in the media, reflection and refraction at media interfaces, and scattering from tissue structure. The attenuation varies depending on the material, and when measured in decibels increases approximately linearly with frequency and depth. Higher frequencies suffer greater attenuation but allow better resolution, so resolution is constrained by the location of an organ with respect to the surface of the body. A rule of thumb for most soft tissue is that the attenuation is roughly 1 dB/cm·MHz. The attenuation is much lower in liquids – in blood, for example, the attenuation is approximately 0.2 dB/cm·MHz. Because the attenuation in bone is very high (~15 dB/cm·MHz), it is an obstacle in ultrasound scans. To compensate for the attenuation in tissue, the gain is increased for more distant echoes. This is known as time gain compensation (TGC), and can be controlled by the equipment operator or set automatically by the scanning equipment.

2.3 Transducers

The transducer used to capture an image has a substantial role in how the image looks, affecting the appearance of features and the shape of the scanned area. For a single-element transducer, the ultrasound beam width at the transducer face is close to the active element width, and diverges as the beam propagates. The beam width stays approximately constant within the near field boundary,

$$\frac{D^2}{4\lambda}, \tag{2.5}$$

where D is the transducer diameter. The lateral resolution is limited by the beam width. The beam can be focussed using a mechanical lens, but this produces optimum resolution only in a narrow focal zone. The beam profile of a single element transducer with and without a focussing lens is illustrated in Figure 2.3. The axial resolution is determined by

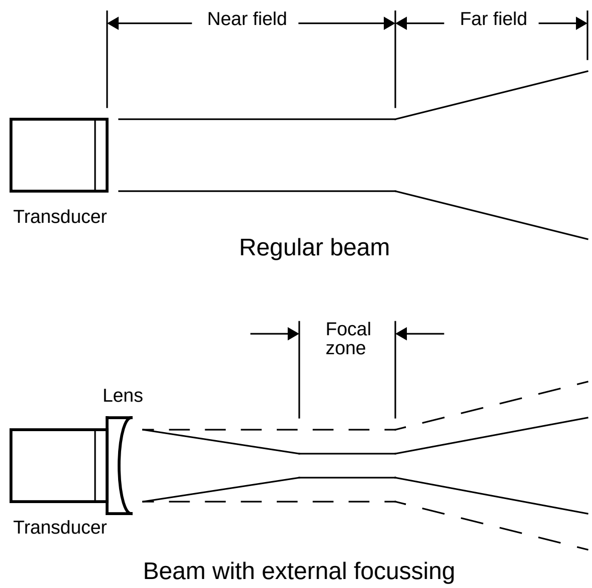


Figure 2.3: Beam profile of a single element transducer

the pulse width, which depends on the mechanical damping in the transducer and the frequency of operation. The appearance of speckle is determined by the transducer resolution, and since axial resolution is typically greater than lateral resolution, speckle spots are elongated in the lateral direction.

Single-element transducers are no longer used. Multi-element arrays, where the beam is synthesized by activating elements at different phases, are used for modern real time B-scanning. A linear sequenced array is composed of a large number of elements arranged side by side along their length and acoustically isolated from each other. The number of elements can range from 20 to 400, but is typically between 128 and 256. The size of each individual element depends on the frequency of operation, and is smaller for higher frequencies. A typical individual element for 3.5 MHz operation is 1 mm wide and 10 mm long. The beam is steered by activating groups of elements in sequence. Because electronic beam focusing is used in the plane of the scan, the number of elements activated at one time depends on the focal depth – more elements are used when the focal depth is greater. The beam is focussed mechanically in the thickness direction (perpendicular to the plane of the scan) by curving the elements or placing an acoustic lens in front of each element. If the linear array has a flat face, the scan is linear and orthogonal to the beam direction. A rectangular image with low distortion, good transverse resolution and high frame rate is produced. The disadvantage of a flat face linear array is that it tends to be large, ranging from 5 to 20 cm wide, which makes it difficult to maintain good contact between the patient and the transducer over a range of orientations. By reducing the number of elements and curving the array face, a wide field of view is maintained with a physically smaller transducer, but a wedge shaped scan is produced. A sector field of 90° is typical for a tightly curved linear sequenced array. A linear phased array is even more compact and also produces a wedge shaped scan. It is also made up of a row of elements, but the beam is steered through an arc by phasing of elements instead of sequencing of elements. Linear phased arrays range from 10 to 30 mm wide, and employ 16 to 128 individual elements. Sector angles are less than 90°, and images exhibit greater distortion because it is difficult to synthesize a consistent beam over the swept range.

2.4 Real time B-scanning

The signal path in a real time B-scanner is shown in the block diagram of Figure 2.4. The analog to digital converter typically has a resolution of 8 bits. When a sector transducer is used, an ultrasound image is generated from signals sampled on a polar coordinate grid. The scan converter, which follows analog to digital conversion, processes the polar coordinate samples to form a raster image. A raster pixel value is interpolated from the closest

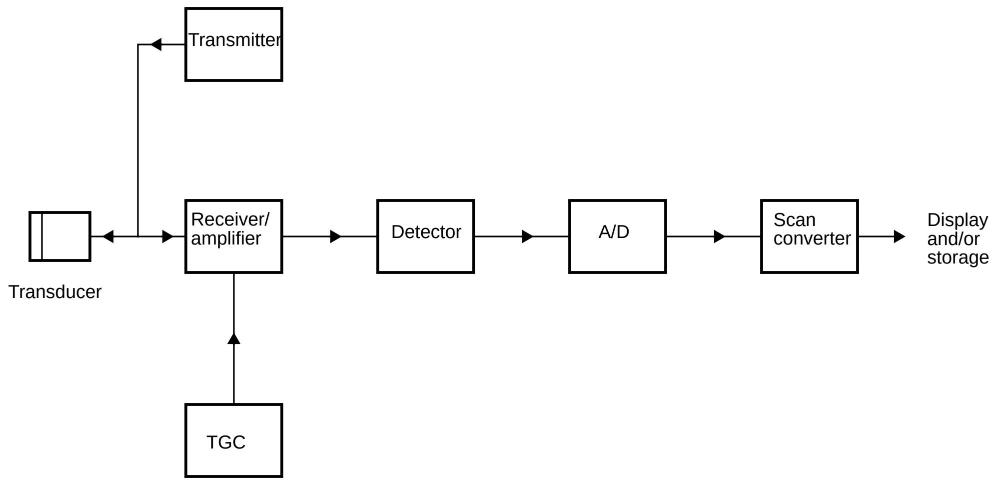


Figure 2.4: Signal path in a real time B-scanner

polar coordinate samples. The scan converted digital image can be stored, or converted back to analog for display on a monitor. Although compression works better on pre-scan converted images [5], they are awkward to work with because they must be scan converted before display. This thesis addresses the compression of scan converted ultrasound images.

The frequency of the transducer selected for a particular exam depends on the location of the organ being examined. High frequency transducers provide more detailed images but can only be used to look at organs close to the surface, for example the thyroid. For abdominal imaging of organs such as the liver, kidneys, or pancreas, use of phased array or curved array transducers at frequencies between 3 and 5 MHz is common.

2.5 Speckle

Speckle is a distinctive feature of ultrasound images that differentiates them from natural images. A simple natural image consists of a few edges against a relatively uniform background, but speckly ultrasound images exhibit texture over the entire ultrasound scanned area. Speckle is one of the features that makes ultrasound images difficult to compress. A brief discussion of speckle, how it is generated and how it is characterized statistically, provides useful background material before looking at ways to compress speckly images better.

Speckle is caused by scattered reflections produced by features that are small with respect to the wavelength. These multiple small reflections result from a rough scattering surface with fine scattering structures. The constructive and destructive interference that occurs when these multiple small reflections with random phase are summed at the transducer produces speckle. The same effect has been observed in the field of optics using lasers, and has been studied extensively in this context [13]. If the scattering structure is smooth enough, the received echoes contain a coherent component which carries some information about the underlying texture. In the case of fully developed speckle, where the scattering surface is very rough, the resulting signal contains no coherent component, and the speckle pattern depends only on the characteristics of the transducer. An example of an organ that produces a particularly speckly ultrasound image is the liver.

The first-order statistics of fully developed ultrasound speckle are studied by Burckhardt [4]. Goodman does the same analysis for laser speckle [24]. The received signal at the transducer is described by a phasor A,

$$A = A_r + jA_i . \tag{2.6}$$

The real and imaginary components of the received signal are shown to be independent Gaussian distributed random variables with identical variance σ^2, so that the joint probability density function (pdf) of A_r and A_i is circular Gaussian. The magnitude of the phasor is used for ultrasound imaging. The pdf of the magnitude v is Rician,

$$f_V(v) = \frac{v}{\sigma^2} \cdot \exp\left(-\frac{v^2 + a^2}{2\sigma^2}\right) \cdot I_0\left(\frac{av}{\sigma^2}\right), \tag{2.7}$$

where

$$a = \sqrt{\mu_{A_r}^2 + \mu_{A_i}^2} \quad (2.8)$$

and μ_x denotes the expected value of x, and I_0 is the modified Bessel function of the first kind of order zero. In the case of fully developed speckle, the real and imaginary components have zero mean, and the pdf of the magnitude is a special case of the Rician distribution, known as the Rayleigh distribution,

$$f_V(v) = \frac{v}{\sigma^2}\exp\left(-\frac{v^2}{2\sigma^2}\right). \quad (2.9)$$

Figure 2.5 shows a number of possible pdfs for the magnitude, including a Rayleigh distribution and a family of Rician distributions with different values of a. Wagner et al. [57] experimentally verify that fully developed speckle shows a good fit with a Rayleigh pdf.

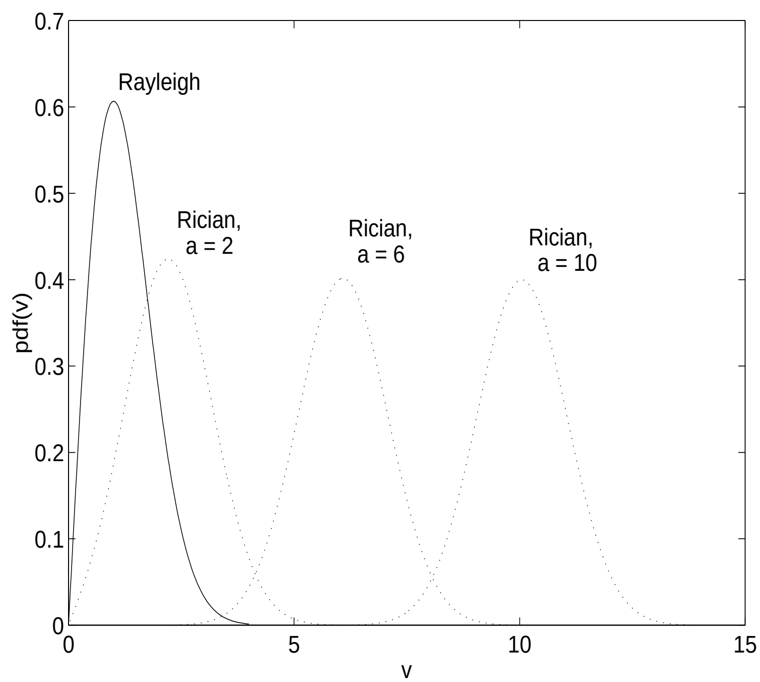


Figure 2.5: Pdf of the phasor magnitude

In the same paper, Wagner et al. also derive and experimentally verify the autocovariance and power spectrum of the received phasor magnitude in the case of fully developed ultrasound speckle. Using their results, the normalized power spectra in the

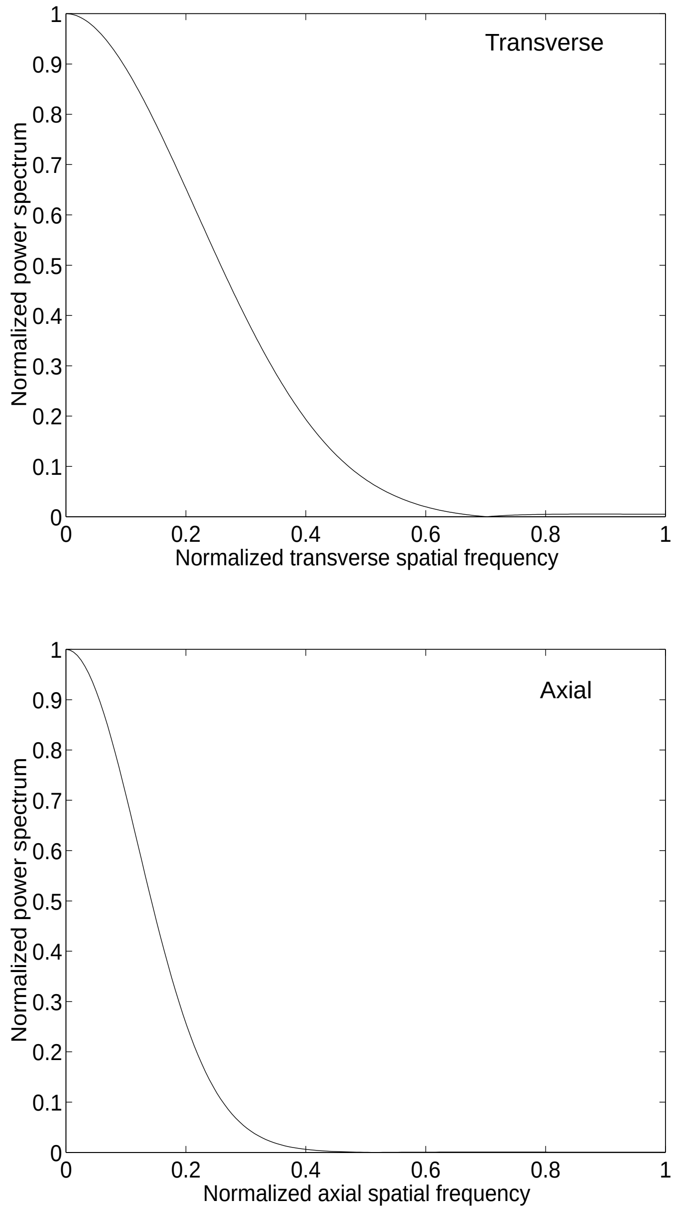


Figure 2.6: Speckle power spectra

transverse and axial directions for a single element rectangular transducer are plotted in Figure 2.6. Note that the power spectra for both directions are lowpass functions. The relative spread of the power spectra for the two different directions cannot be compared directly in the figure, since transverse position is normalized relative to transducer dimension, but axial position is normalized relative to pulse width.

2.6 Summary

Medical ultrasound imaging uses reflections from pressure waves to generate an image. As the mechanical ultrasound signal propagates, it is reflected, refracted, and attenuated; these phenomena all need to be accounted for in order to create an image. The transducer is an important part of an ultrasound imaging system, and has a significant effect on the appearance of an image. Multi-element transducers are currently in widespread use, and allow real-time ultrasound imaging. Speckle is a distinguishing feature of ultrasound images, which results from scattered reflections produced by rough scattering surfaces with fine scattering structures with respect to wavelength. In the frequency domain, speckle exhibits a low pass characteristic.

Chapter 3

Image Compression and Wavelets

3.1 Introduction

A monochrome image is composed of an array of pixels. Each pixel is a binary integer representing the brightness at a location. In the original image, all the pixels have the same precision and approximately the same statistics. A lossy compression system generates a representation of the original image, from which an estimate of the original is recovered. The system has two conflicting goals: to minimize the number of bits used to represent the image, and to maximize the fidelity of the recovered estimate with respect to the original. Transform coding and subband coding achieve compression by transforming the original pixels into a set of coefficients with differing statistics, and allocating bits to coefficients depending on their statistics. Quantization is used to map an infinite continuous set of real transformed values to a finite discrete set of real values, and determines the number of bits allocated to represent each transformed coefficient. Scalar quantization is a simple technique that quantizes coefficients on an individual basis; scalar quantization followed by entropy coding produces a compact lossy representation of the transformed coefficients. Wavelet analysis is a relatively new mathematical tool that is closely related to subband coding, and insight from this relationship has led to improved lossy image compression systems.

3.1.1 Mathematical concepts

The discussion of coding schemes and wavelets in the following sections require the definition of some mathematical concepts. Let Z denote the set of integers. The inner product of two discrete real-valued time sequences $x[n]$ and $y[n]$ over Z is

$$\langle x, y \rangle = \sum_{n=-\infty}^{\infty} x[n]\, y[n] \,. \tag{3.1}$$

Let R denote the set of real numbers. The inner product of two continuous real-valued time functions $f(t)$ and $g(t)$ over R is

$$\langle f, g \rangle = \int_{-\infty}^{\infty} f(t) g(t) dt \,. \tag{3.2}$$

The norm of a function is defined in terms of the inner product as

$$\|x\| = \sqrt{\langle x, x \rangle} \,. \tag{3.3}$$

A discrete time function belongs to the space of square summable sequences,

$$x[n] \in l_2(Z) \,, \tag{3.4}$$

if

$$\|x\| < \infty \,. \tag{3.5}$$

A continuous time function belongs to the space of square integrable functions,

$$f(t) \in L_2(R) \tag{3.6}$$

if

$$\|f\| < \infty \,. \tag{3.7}$$

3.1.2 Fidelity measures

To quantify the performance of a lossy image compression system, a metric comparing the recovered estimate to the original image is required. An objective numerical measure is useful because it provides a neat summary of image quality, and allows convenient comparisons based on a consistent definition. Numerical measures of image quality based on squared error are common.

The squared error between a signal x and its estimate $\hat{x}$ is defined as

$$|x - \hat{x}|^2 . \tag{3.8}$$

The mean squared error (MSE) is defined as

$$MSE = E\{|x - \hat{x}|^2\} . \tag{3.9}$$

The peak signal to noise ratio (PSNR) can be viewed as a normalized mean squared error reported using a logarithmic scale, and is defined as

$$PSNR = 10\log_{10}\frac{M^2}{MSE}, \tag{3.10}$$

where M is the maximum peak-to-peak value of the signal. For image compression,

$$MSE = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}|x_{mn} - \hat{x}_{mn}|^2 \tag{3.11}$$

for an $M \times N$ pixel grayscale image. For an 8-bit grayscale image,

$$M = 255 . \tag{3.12}$$

Peak signal to noise ratio will be used throughout this thesis as an objective numerical measure of compressed image quality.

Mean squared error is commonly used as a measure of fidelity because of its mathematical tractability; however, it is not the best metric for assessing perceptual quality. Mean squared error is popular because it is easy to calculate and manipulate. When a transformation is orthogonal, the squared error is the same in the transform domain and the signal domain. In practice, it is common to employ orthogonal or nearly orthogonal transforms, and use of a mean squared error metric means that the error can be computed in the transform domain, without performing an inverse transform. The drawback to using means squared error is that it is only a mediocre measure of subjective quality. The advantage of being a summary statistic for the entire image is a disadvantage when measuring perceptual quality. Errors concentrated in a spatial region can produce visible and annoying distortion in that region, but result in only a small MSE. Errors concentrated in frequency bands to which the human visual system are particularly sensitive can also produce perceptual distortion that is not proportionally reported with a simple mean squared error metric. In the particular case of medical image compression, where the goal is to maintain diagnostic integrity, assessment of diagnostic features by expert viewers is the most relevant way to judge image quality.

3.2 Entropy coding

Entropy is a measure of information [44]. Consider a discrete random variable X, with a finite alphabet of size N, $\{x_0, x_1, \ldots, x_{N-1}\}$. The random variable X is characterized by its probability mass function,

$$p_X(x_i) = \text{prob}(X = x_i) \qquad i = 0, 1, \ldots, N-1 . \tag{3.13}$$

The zeroth-order entropy of the random variable X, $H(X)$, is defined as

$$H(X) = -\sum_{i=0}^{N-1} p_X(x_i) \log p_X(x_i) . \tag{3.14}$$

Using the base two logarithm, entropy is measured in bits.

Entropy coding refers to the use of a variable length code to losslessly represent a sequence of symbols from a discrete alphabet. The term entropy code comes from the fact that the entropy provides a minimum bound for the average codeword length in the case of a memoryless source. A practical entropy code must be uniquely decodeable, so that there is only one possible sequence of codewords for any unique input sequence. A prefix code, where no codeword is allowed to be a prefix of any other codeword, is uniquely decodeable. Prefix codes are also referred to as instantaneous codes, because a codeword can be decoded unambiguously upon reception, without receiving any other codewords.

The following sections discuss three entropy coding techniques. Huffman coding is commonly encountered, and is used in most JPEG implementations. Arithmetic coding and stack-run coding are reviewed in some detail, because they are the two coding schemes used in the experiments with space-frequency segmentation that are reported in Chapter 6.

3.2.1 Huffman coding

Huffman coding [28] is a simple entropy coding scheme that produces an optimal binary prefix code. When symbol probabilities are powers of $1/2$, the average length of a binary Huffman code achieves the entropy bound. At each stage in the generation of a binary Huffman code, the two lowest probability symbols are grouped, assigning a “0” to one of the symbols, and a “1” to the other symbol. The process is successively repeated, resulting in a binary tree with leaves which correspond to the original symbols, and branches which are labelled with a “0” or a “1”. A codeword is formed by reading the “0” or “1” from the branches as the tree is traversed from the root node to the leaf corresponding to that symbol. An example of Huffman coding is shown in Figure 3.1. Note that low probability symbols are assigned longer codewords, while high probability symbols are assigned shorter codewords.

Huffman coding is simple and generally works well; however, it has some drawbacks. Because each symbol is assigned a codeword with an integral number of bits, the

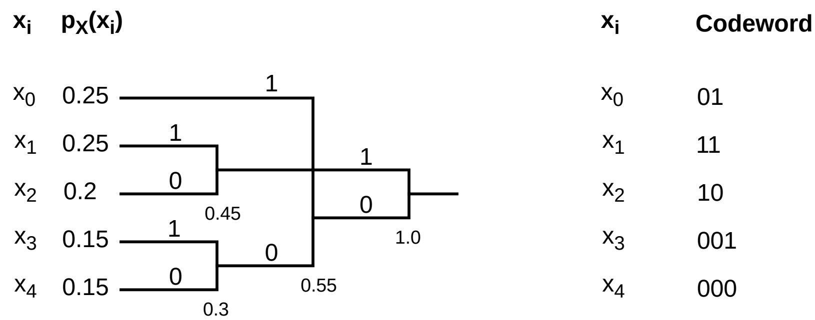


Figure 3.1: Example of Huffman coding

average length of a Huffman code achieves the entropy bound only when symbol probabilities are powers of $\frac{1}{2}$. Grouping symbols into meta-symbols for codeword assignment addresses the integral bit problem, but the accompanying increase in alphabet size makes the coding and decoding task more complicated. The code table needs to be sent as side information, and when many sources with different statistics need to be coded (as is the case with space-frequency segmentation), the amount of side information can be considerable. Since a pass to gather statistics must be made before assigning codewords to symbols, Huffman coding is not by nature adaptive. Huffman coding can be made adaptive, but at the expense of increased complexity.

3.2.2 Arithmetic coding

An entropy coding technique that addresses some of the problems with Huffman coding is arithmetic coding. In Huffman coding, grouping symbols into meta-symbols removes the integral bit restriction for individual symbol codewords, producing a more efficient code. Arithmetic coding [2] performs an ultimate grouping, effectively considering the entire sequence as a single symbol. Grouping so many symbols together appears to be impractical, since the alphabet of possible codewords becomes very large, but arithmetic coding solves this problem by encoding and decoding incrementally. Incremental coding means that arithmetic coding can easily be made adaptive, and no side information describing source statistics needs to be sent.

Arithmetic coding represents a sequence of symbols as an interval in the range [0, 1). The size of the interval is the probability of the sequence, and the location of the interval uniquely identifies the sequence. More probable sequences are represented with a larger interval specified by less bits, and less probable sequences are represented with a smaller interval specified by more bits. In theory the interval is represented by real numbers, but a practical coder uses a binary fraction with limited precision, accommodating long sequences that exceed the precision by rescaling the interval when required. Incremental coding occurs because bits specifying the interval are sent in order of precision. When a string of symbols must lie within the current interval, it can be decoded without receiving any further bits.

Arithmetic coder operation is illustrated with an example. Consider a source with a four symbol alphabet, denoted {*a*, *b*, *c*, *d*}, with symbol probabilities as defined in Table 3.1. In the unit interval [0, 1), each symbol is represented by an interval whose size is equal to the symbol probability. The intervals are also shown in Table 3.1. Assume that the

Table 3.1: Symbol probabilities and intervals in the unit interval

Symbol	**Probability**	**Interval**
a	0.6	[0, 0.6)
b	0.1	[0.6, 0.7)
c	0.2	[0.7, 0.9)
d	0.1	[0.9, 1.0)

sequence *abac* is to be coded. The first symbol, *a*, is represented by the interval [0, 0.6). The next interval is defined by subdividing [0, 0.6) in proportion to the probability of the next symbol *b*, according to the interval assignments defined in Table 3.1. Instead of [0.6, 0.7) with respect to the unit interval, the next interval is $[0.6 \times 0.6,\ 0.7 \times 0.6)$.Note that the size of the interval [0.36, 0.42), is the cumulative probability of the two symbols *ab*. The process continues for successive symbols, so that the sequence *abac* is represented by the final interval [0.3852, 0.3924). The intervals are shown in Figure 3.2, where the size of the interval in the figure has been scaled after each symbol so that small intervals remain visible.

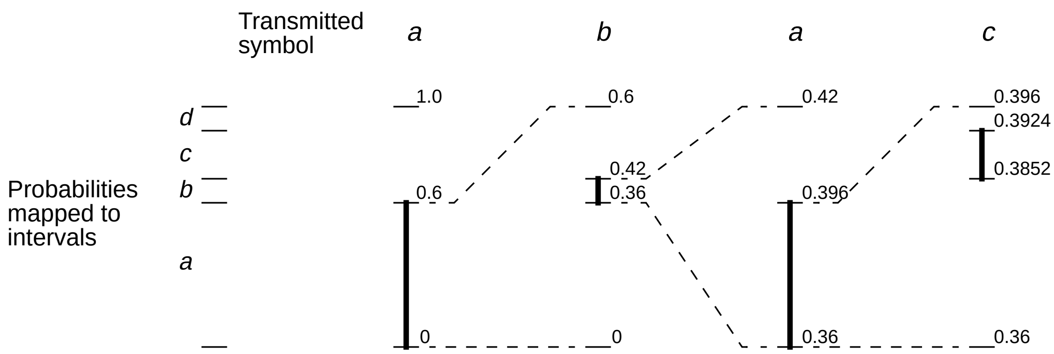


Figure 3.2: Intervals for the sequence *abac*

The interval representing a sequence is coded as a string of bits that define the largest binary interval contained within the symbol interval. Bits are transmitted in order of precision, most significant bit first. The rule for refining the binary interval is that a zero signifies the bottom half and a one signifies the top half of the current range. As the binary code string is generated, the incremental nature of the encoding process becomes apparent. After each symbol, the bits that define the smallest binary interval that contains the current symbol interval can be sent. In the example, the first interval [0, 0.6) straddles 0.5, the midpoint of the current known binary interval [0, 1), so no bits are transmitted for the symbol *a*. After the second symbol *b*, the smallest binary interval that contains the code interval [0.36, 0.42) is [0.25, 0.5), so the bits 01 are transmitted. The next symbol interval [0.36, 0.396) straddles 0.375, the midpoint of the current binary range [0.25, 0.5), so again no bits are transmitted for the symbol *a*. The final symbol interval [0.3852, 0.3924) results in the transmitted bits 100. A few bits are required to terminate the message by refining the binary interval so that it is within the symbol interval. For the example, the bits 011 are required. The final binary code string 01100011 represents the binary interval

[0.38671875, 0.390625). Table 3.2 describes the encoding process, showing the transmitted bits (excluding the message termination bits) and the intervals that they represent.

Table 3.2: Encoding the sequence *abac*

Symbol	**Cumulative symbol interval**	**Cumulative binary interval**	**Tx bit**
a	[0, 0.6)	[0, 1)	-
b	[0.36, 0.42)	[0, 0.5)	0
		[0.25, 0.5)	1
a	[0.36, 0.396)	[0.25, 0.5)	-
c	[0.3852, 0.3924)	[0.375, 0.5)	1
		[0.375, 0.4375)	0
		[0.375, 0.40625)	0

The decoding process refines a binary interval from the received bits, incrementally decoding symbols when the binary interval is within the symbol interval. The first received bit, 0, specifies the interval [0, 0.5). Since [0, 0.5) lies completely within [0, 0.6), the first symbol must be *a*. After receiving 1100, the binary interval [0.375, 0.40625) is within the symbol interval for the sequence *ab*, so the next symbol *b* is decoded. The decoding process for the entire example sequence is shown in Table 3.3.

Table 3.3: Decoding the sequence *abac*

Rx bit	**Cumulative binary interval**	**Decoded symbol**
0	[0, 0.5)	*a*
1	[0.25, 0.5)	
1	[0.375, 0.5)	
0	[0.375, 0.4375)	
0	[0.375, 0.40625)	*b*
0	[0.375, 0.390625)	*a*
1	[0.3828125, 0.390625)	
1	[0.38671875, 0.390625)	*c*

Factors that make arithmetic coding effective are apparent even in this short example. In both occurrences of the high probability symbol *a*, no bits are generated at the transmitter and only a single bit is required at the receiver to decode the symbol. More bits are required to represent and decode the symbol *c*, and the lowest probability symbol *b* generates the most bits at the transmitter and requires the most bits before being decoded at the receiver. Excluding the three bits required for termination, the four symbol sequence is coded with 5 bits, which is less than the 8 bits required by a fixed length code.

In a practical implementation, the symbol range is represented by fixed precision rather than floating point numbers. The practical coder must ensure that the quantities representing the high and low interval limits do not underflow as the interval becomes small.

Since source statistics are likely unavailable in the real world, the practical coder generates a histogram by counting symbol occurrences as it codes. As long as the encoder and decoder start with the same initial histogram, no explicit information on source statistics is required. Use of a histogram generated in this way allows limited adaptation to changing statistics, and permits more sophisticated adaptation techniques to be applied easily. One method to increase the adaptation is to modify the histogram generation so that recently encountered symbols dominate. Another technique to improve performance is to allow multiple histograms, and to choose the histogram used to code a symbol based on that symbol's context. The arithmetic coder used for experiments in this thesis uses a single histogram, and each symbol simply increments the bin count for that symbol by one.

3.2.3 Stack-run coding

Stack run coding, proposed by Tsai, Villasenor and Chen [51], is a simple but effective scheme for encoding quantized subband image coefficients. The coefficients are represented using a four symbol alphabet, then entropy coded using an adaptive arithmetic coder. The small symbol alphabet allows the arithmetic coder to adapt quickly. After subband decomposition, there are many zero-valued coefficients in the high pass bands, and often these zeros occur together. Run-length coding of consecutive zeros is key to efficient stack-run coding. Effective stack-run coding also depends on the energy compaction effect of subband decomposition. Energy compaction produces a large number of small magnitude coefficients in the high pass subbands. Since stack-run codes a coefficient on a bit by bit basis, small magnitude coefficients are coded compactly. If coefficient magnitudes are large, other coding techniques, which code an entire coefficient using a single symbol, are better at representing non-zero coefficients.

Encoding begins by converting the two-dimensional image into a one-dimensional sequence. Tsai, Villasenor and Chen propose a raster scan, but Raffy [38] suggests scanning back and forth to better exploit correlation at image edges. The name stack-run coding comes from the distinction between zero and non-zero coefficients. The bits of a non-zero coefficient form a “stack”. A string of consecutive zeros forms a “run”, and is represented by the number of zeros in the run. Stacks and runs are encoded using the alphabet shown in Table 3.4.

Table 3.4: The stack-run alphabet

Symbol	**Context**	**Value represented**
“0”	stack	binary zero in the coefficient
“1”	stack	binary one in the coefficient
“+”	stack	MSB of a positive coefficient
	run	binary one in the run length
“-”	stack	MSB of a negative coefficient
	run	binary zero in the run length

The group of symbols representing a stack or run is transmitted least significant bit first. Since it is always one, the most significant bit does not need to be transmitted when there is a way to implicitly terminate stacks and runs. A run is always positive, and in most cases the most significant bit can be dropped. No symbols are produced when the most significant bit of a run of length one is dropped, so the most significant bit is kept for all runs of length $2^n - 1$. A transition from “+” or “-” to 1 or 0 terminates a run, when the most significant retained bit of the run encounters the least significant bit of the following stack. Coefficients can be positive or negative, so the most significant bit of a stack is represented by its sign, and a transition from “0” or “1” to “+” or “-” terminates the stack. Dropping the most significant bit of a stack means that a coefficient with value +1 or -1 cannot be distinguished from a run, but incrementing all stacks by one solves the problem by ensuring a stack of value +1 or -1 is never encountered.

The symbols are encoded using an adaptive arithmetic coder. For better performance, two sets of probability tables are used, both containing entries for all four symbols. Runs and the least significant bit of stacks are coded together, and their probability table is dominated by entries for “+” and “-” symbols. The rest of the stack symbols are coded together, and their probability table is dominated by entries for “0” and “1”.

An example of stack-run coding (from Raffy) is shown in Figure 3.3. Note that the most significant bit is retained for the first run of seven zeros.

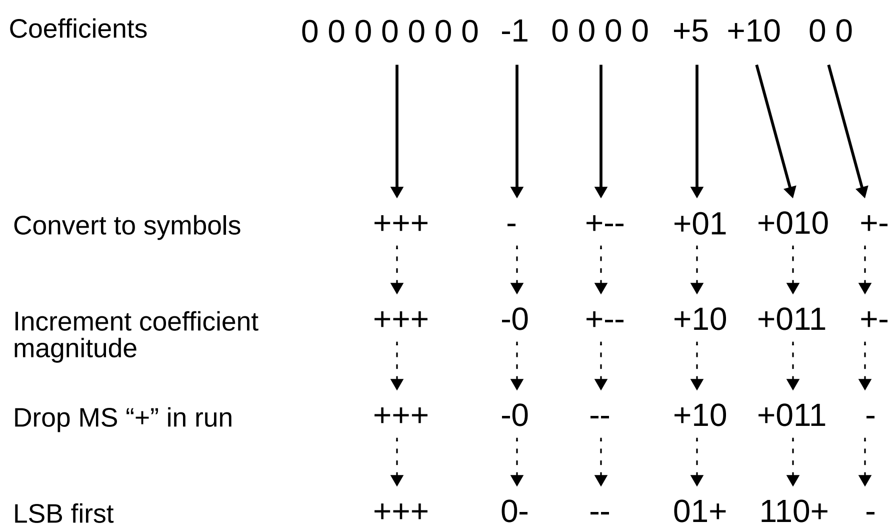


Figure 3.3: Stack-run coding example

3.3 Scalar quantization

Scalar quantization is simply the mapping of an input symbol to a quantization index. When the input symbol alphabet is larger than the quantization index alphabet, there is compression of information, but also loss of information. All symbols within a range of values are mapped to the same quantization index; this range defines the quantization cell for that index. The inverse quantization operation maps the quantization index to an output symbol. Typically, scalar quantization maps an infinite set of real inputs to a non-infinite set of integer indexes, and inverse quantization maps the non-infinite set of integer indexes to a non-infinite set of real outputs. The difference between the input symbol and the output symbol is the quantization error for that symbol. A scalar quantizer is termed regular if the individual cells are contiguous, the ordered cells span a contiguous range, there is no overlap between cells, and the output value of a cell lies within that cell. When mapping a real symbol alphabet using a regular quantizer with a finite index alphabet, one of the cells will have a lower range of $-\infty$ and one of the cells will have an upper range of ∞; these two cells are called overload cells. All other cells are called granular cells. A uniform quantizer is a regular quantizer where all granular cells are the same size, and the output value for each granular cell is the midpoint of the cell. A uniform quantizer is completely defined by its alphabet size, cell size or step size, and the range covered by the granular cells.

Quantization is a large topic and has been studied extensively by the source coding community. Gersho and Gray [23] is a good reference for quantization and compression, and also discusses entropy coding, transform coding, and subband coding. The performance of uniform scalar quantization followed by entropy coding approaches that of optimum scalar quantization [20]; this combined with its simplicity has made it a popular choice for image compression systems. Vector quantization, which codes groups of symbols rather than individual symbols, has also been applied successfully to image compression. In the most effective coders, whether scalar or vector quantization is used, the entropy coding technique, quantization and transformation work together to produce good compression performance. The choice of quantization method and how it is applied to the coefficients is intimately connected to how the coefficients are generated.

3.4 Transform coding

In transform coding [8], a linear transformation is applied to the original signal. The transformed signal is quantized, then the reverse transformation is applied to generate an estimate of the original. A block diagram of a transform coding system is shown in Figure 3.4. For efficient implementation, a block transform divides the signal into regular blocks, and applies the transformation on a block-by-block basis.

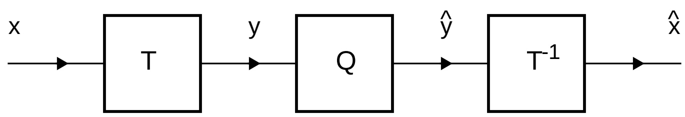


Figure 3.4: Transform coding system

In a one-dimensional block transform system, a block of the original signal can be described as a column vector,

$$\underline{x} = \begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ x_N \end{bmatrix}, \tag{3.15}$$

where N is the block size. The transform coefficients of the block are obtained by multiplying by the $N \times N$ transformation matrix $\boldsymbol{T}$,

$$\underline{y} = \boldsymbol{T}\underline{x}. \tag{3.16}$$

The transform coefficients can be viewed as the projection of the signal onto the basis composed of the rows of $\boldsymbol{T}$. The inverse transform is applied to the transform coefficients to recover the original signal exactly,

$$\underline{x} = \boldsymbol{T}^{-1}\underline{y}. \tag{3.17}$$

When the inverse transform is applied to the quantized transform coefficients $\hat{\underline{y}}$, the result is an estimate of the original signal,

$$\hat{\underline{x}} = \boldsymbol{T}^{-1}\hat{\underline{y}}. \tag{3.18}$$

If the transformation is unitary,

$$\boldsymbol{T}^{*T}\boldsymbol{T} = \boldsymbol{I}, \tag{3.19}$$

where $\boldsymbol{I}$ is the identity matrix, then

$$\|\underline{y} - \hat{\underline{y}}\|^2 = \|\underline{x} - \hat{\underline{x}}\|^2. \tag{3.20}$$

This property of unitary transformations is useful because it means that the quantization error in the transform domain and the signal domain are the same, when a squared error metric is used.

In two dimensions, the inputs and outputs of the block transform coding system can be described by matrices. If the transform is separable, which is the usual case, it can be applied independently to the rows and columns

$$\boldsymbol{Y} = \boldsymbol{T}\boldsymbol{X}\boldsymbol{T}^T, \tag{3.21}$$

and if the transform is also unitary then

$$\boldsymbol{X} = \boldsymbol{T}^{*T}\boldsymbol{Y}\boldsymbol{T}^*. \tag{3.22}$$

Transform coding achieves compression by reducing the correlation in the transform coefficients. Decorrelation removes redundant information, allowing a more compact representation. The transformation also results in energy compaction. This change in the energy distribution of the coefficients produces a small group of transform coefficients with large variance and a large group of coefficients with small variance. The coefficients with large variance can be quantized with a larger proportion of the bit budget, resulting in an efficient representation.

Viewing the transformation as a projection onto a basis, it is clear that the choice of basis vectors determines the decorrelating and energy compacting properties of the transformation. The optimal decorrelating transformation is the Karhunen-Loève Transform

(KLT), which produces pairwise uncorrelated transform coefficients. The basis vectors of the KLT are the eigenvectors of the autocorrelation matrix,

$$\boldsymbol{T} = \boldsymbol{U}^T, \tag{3.23}$$

where

$$\boldsymbol{U} = \left[\underline{u}_1 \ \underline{u}_2 \ \dots \ \underline{u}_k\right], \tag{3.24}$$

and $\underline{u}_i$ is the i$^{\text{th}}$ eigenvector and λ_i is the i$^{\text{th}}$ eigenvalue

$$\boldsymbol{R_X}\underline{u}_i = \lambda_i \underline{u}_i, \tag{3.25}$$

and $\boldsymbol{R_X}$ is the autocorrelation matrix

$$\boldsymbol{R_X} = E\{\underline{x}\underline{x}^T\}. \tag{3.26}$$

The KLT is signal dependent and awkward to compute when the block size is large. A more practical alternative is the Discrete Cosine Transform (DCT), which is equivalent to the Discrete Fourier Transform of a block using symmetric extension. The DCT generally produces transform coefficients with low correlation when applied to natural images. The elements of the first row of the DCT transformation matrix are all equal to $\sqrt{1/N}$. The other rows are defined by

$$t_{jk} = \sqrt{\frac{2}{N}}\cos\left(\frac{\pi}{N}(j-1)\left[(k-1)+\frac{1}{2}\right]\right) \qquad j = 2, 3, \dots, N; k = 1, 2, \dots, N \tag{3.27}$$

where t_{jk} is the element of the j$^{\text{th}}$ row and k$^{\text{th}}$ column.

A well known transform coder for grayscale and color images is defined by the JPEG standard [37]. The JPEG coder uses the DCT followed by scalar quantization with a choice of step sizes. The DCT basis concentrates most of the energy in a block into a single coefficient, termed the DC coefficient. DC coefficients are coded by encoding the difference with respect to the DC coefficient from the previous block. The other coefficients in a block are encoded using a combination of Huffman coding and run-length coding, where a string of consecutive values is represented by the number of values.

3.5 Subband coding and filter banks

Subband coding [58] [59] is a source coding technique where the frequency domain characteristics of the signal are exploited to code it efficiently. The encoder separates the original signal into a set of frequency subbands by filtering it with a set of analysis filters. To maintain critical sampling, the analysis filter outputs are subsampled to produce the subband coefficients. The subband coefficients are quantized to produce a lossy representation of the original signal. To generate the estimate of the signal from the quantized subband coefficients, the coefficients are interpolated, filtered by synthesis filters, and summed. In the absence of quantization, the analysis followed by synthesis is lossless, even when implemented with realizable filters.

Perfect reconstruction

The two channel perfect reconstruction filter bank [55], illustrated in Figure 3.5, is at the heart of a subband coding system. In a one-dimensional system, the analysis filters h_0 and

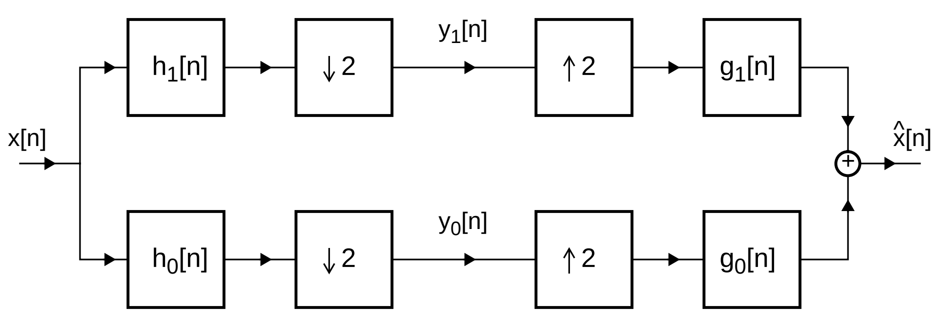


Figure 3.5: One-dimensional two channel perfect reconstruction filter bank

h_1 split the input sequence x into two subbands. Subsampling the analysis filter outputs by two maintains critical sampling, and produces two sequences y_0 and y_1, each with half the number of elements as the original sequence. To generate an estimate $\hat{x}$ of the original signal, the filtered sequences are upsampled by a factor of two, filtered by the synthesis filters g_0 and g_1, and summed. The analysis and synthesis filters are chosen so that aliasing effects cancel, making the estimate a perfect reconstruction in the absence of quantization. In the z-transform domain, the estimate can be written in terms of the input and filter responses as

$$\hat{X}(z) = \frac{1}{2}[H_0(z)G_0(z) + H_1(z)G_1(z)]X(z) + \frac{1}{2}[H_0(-z)G_0(z) + H_1(-z)G_1(z)]X(-z). \quad (3.28)$$

To obtain perfect reconstruction,

$$\hat{X}(z) = X(z), \quad (3.29)$$

all the terms with $X(z)$ as a factor must sum to $X(z)$,

$$H_0(z)G_0(z) + H_1(z)G_1(z) = 2, \quad (3.30)$$

and all the aliasing terms which contain the factor $X(-z)$ must cancel,

$$H_0(-z)G_0(z) + H_1(-z)G_1(z) = 0. \quad (3.31)$$

Equation 3.30 and Equation 3.31 can be written in matrix form as

$$\boldsymbol{H}_{\boldsymbol{m}}^T(z)\begin{bmatrix} G_0(z) \\ G_1(z) \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \quad (3.32)$$

where $\boldsymbol{H}_{\boldsymbol{m}}(z)$ is the analysis modulation matrix

$$\boldsymbol{H}_{\boldsymbol{m}}(z) = \begin{bmatrix} H_0(z) & H_0(-z) \\ H_1(z) & H_1(-z) \end{bmatrix}. \quad (3.33)$$

Assuming that $\left(\boldsymbol{H}_{\boldsymbol{m}}^T(z)\right)^{-1}$ exists, Equation 3.32 can be rewritten as

$$\begin{bmatrix} G_0(z) \\ G_1(z) \end{bmatrix} = \frac{2}{\det(\boldsymbol{H_m}(z))} \begin{bmatrix} H_1(-z) \\ -H_0(-z) \end{bmatrix}. \tag{3.34}$$

In practical systems it is often desirable that the filters have a finite impulse response (FIR) characteristic. In order to achieve perfect reconstruction with FIR filters, the determinant of the analysis modulation matrix must satisfy the condition

$$\det(\boldsymbol{H_m}(z)) = \alpha z^{-2k-1}, \tag{3.35}$$

where $\alpha = 2$ for perfect reconstruction[1]. Substituting Equation 3.35 into Equation 3.34 defines the synthesis filters in terms of the analysis filters,

$$G_0(z) = z^{2k+1} H_1(-z) \tag{3.36}$$

and

$$G_1(z) = -z^{2k+1} H_0(-z). \tag{3.37}$$

According to Equation 3.36 and Equation 3.37, the synthesis filters will be non-causal if the analysis filters are causal. The synthesis filters can be made causal by multiplying by z^{-2k-1}, so that Equation 3.29 becomes

$$\hat{X}(z) = z^{-2k-1} X(z), \tag{3.38}$$

describing perfect reconstruction with delay $2k + 1$.

Finding solutions that satisfy Equation 3.30 and Equation 3.31 is key to designing effective subband coders, and extensive research has been devoted to this problem [53]. One approach is the Quadrature Mirror Filter (QMF) bank [52]. A QMF bank is specified by three equations:

$$H_1(z) = H_0(-z), \tag{3.39}$$

---

1. For a proof see Vetterli and Kovačević [55] Section 3.2.2.

$$G_0(z) = H_0(z), \tag{3.40}$$

and

$$G_1(z) = -H_1(z). \tag{3.41}$$

By convention, H_0 is a lowpass filter and H_1 is a highpass filter. Equation 3.39 means that in the frequency domain the highpass analysis filter is the mirror image of the lowpass analysis filter about $\pi/2$; this fact gives the Quadrature Mirror Filter bank its name. Substituting Equation 3.39, Equation 3.40 and Equation 3.41 into Equation 3.30 results in

$$H_0^2(z) - H_0^2(-z) = 2. \tag{3.42}$$

The problem with QMFs is that there is no non-trivial FIR solution to Equation 3.42, although practical filters can be designed that result in almost perfect reconstruction. Smith and Barnwell [48] solve the problem of perfect reconstruction using FIR filters by choosing

$$H_1(z) = -H_0(-z^{-1})z^{-2k-1}, \tag{3.43}$$

leading to what they call Conjugate Quadrature Filters (CQFs). Equation 3.36, Equation 3.37 and Equation 3.43 are substituted into Equation 3.30, resulting in

$$H_0(z)H_0(z^{-1}) + H_0(-z)H_0(-z^{-1}) = 2 \tag{3.44}$$

The filter bank is designed by solving Equation 3.44 using spectral factorization [53].

Basis vector interpretation

Viewing perfect reconstruction filter banks from the perspective of a linear combination of basis vectors provides useful insight into the way filter banks work. This basis vector perspective of filter banks is closely related to wavelet analysis, and is particularly important for understanding space-frequency segmentation.

An orthonormal series expansion of the square summable sequence $x[n]$ is defined as

$$x[n] = \sum_{k \in z} X[k]\, \varphi_k[n] , \tag{3.45}$$

where

$$X[k] = \langle \varphi_k[l], x[l] \rangle \tag{3.46}$$

are the transform coefficients with respect to $\{\varphi_k[n]\}$. $\{\varphi_k[n]\}$ is an orthonormal basis for $l_2(Z)$, and satisfies the orthogonality condition

$$\langle \varphi_k[n], \varphi_l[n] \rangle = \delta[k-l] . \tag{3.47}$$

Conservation of energy,

$$\|x\|^2 = \|X\|^2 , \tag{3.48}$$

is an important property of orthonormal expansions.

A biorthogonal series expansion allows different bases to be used for analysis and synthesis,

$$x[n] = \sum_{k \in z} \tilde{X}[k]\, \tilde{\varphi}_k[n] = \sum_{k \in z} X[k]\, \varphi_k[n] , \tag{3.49}$$

where

$$\tilde{X}[k] = \langle \varphi_k[l], x[l] \rangle \tag{3.50}$$

and

$$X[k] = \langle \tilde{\varphi}_k[l], x[l] \rangle , \tag{3.51}$$

and $\{\tilde{\varphi}_k[l]\}$ and $\{\varphi_k[l]\}$ are dual bases that satisfy the orthogonality constraint

$$\langle \varphi_k[n], \tilde{\varphi}_l[n] \rangle = \delta[k-l] . \tag{3.52}$$

Separating the analysis and synthesis bases makes a biorthogonal system less restrictive than an orthogonal system. The characteristics of the analysis and synthesis basis can be made quite different, and can be tailored to the analysis or synthesis task. Since the bases are related to filters in a filter bank, biorthogonality means that the analysis and synthesis filters can be designed with different characteristics, while still forming a perfect reconstruction system. Conservation of energy for a biorthogonal expansion is expressed as

$$\|x\|^2 = \langle X[k], \tilde{X}[k] \rangle . \tag{3.53}$$

Equation 3.53 means that the energy of the transform coefficients is no longer equal to the energy of the original signal. The impact for subband coding is that the squared error in the transform domain is no longer equivalent to the squared error in the signal domain. In practice, biorthogonal expansions that are almost orthogonal are usually used, so that the convenience of equating squared error in the transform and signal domain can be maintained with little penalty in terms of accuracy.

A perfect reconstruction filter bank, described by Figure 3.5 and Equation 3.30 and Equation 3.31, implements a biorthogonal series expansion[2]. The analysis basis functions are time reversed even shifts of the analysis filter impulse responses,

$$\tilde{\varphi}_{2k}[l] = h_0[2k - l] \tag{3.54}$$

and

$$\tilde{\varphi}_{2k+1}[l] = h_1[2k - l] , \tag{3.55}$$

and the synthesis basis functions are even shifts of the synthesis filter impulse responses,

$$\varphi_{2k}[n] = g_0[n - 2k] \tag{3.56}$$

and

$$\varphi_{2k+1}[n] = g_1[n - 2k] . \tag{3.57}$$

---

2. For a proof see Vetterli and Kovačević [55] Section 3.2.1.

The transform coefficients are the subsampled outputs of the analysis filters,

$$X[2k] = y_0[k] \tag{3.58}$$

and

$$X[2k+1] = y_1[k]\ . \tag{3.59}$$

Because of the relationship between perfect reconstruction and biorthogonality, the filters in a perfect reconstruction filter bank are referred to as biorthogonal filters.

A perfect reconstruction system constrained to implement an orthonormal series expansion results in orthogonal filters. Equating the analysis and synthesis bases relates the analysis and synthesis filters,

$$G_0(z) = H_0(z^{-1}) \tag{3.60}$$

and

$$G_1(z) = H_1(z^{-1})\ . \tag{3.61}$$

Writing Equation 3.30 in terms of G_0 using Equation 3.60 and Equation 3.61 results in an expression describing the synthesis lowpass filter,

$$G_0(z)G_0(z^{-1}) + G_0(-z)G_0(-z^{-1}) = 2\ . \tag{3.62}$$

Equation 3.62 can also be written in terms of the synthesis highpass filter as

$$G_1(z)G_1(z^{-1}) + G_1(-z)G_1(-z^{-1}) = 2 \tag{3.63}$$

The relationship between the lowpass and highpass synthesis filters is established by applying orthogonality conditions to Equation 3.62 and Equation 3.63 and solving, resulting in

$$G_1(z) = -z^{-2K+1}G_0(-z^{-1})\ .\text{[3]} \tag{3.64}$$

The filters in an orthogonal filter bank are derived from a single prototype filter; in this case all the filters have been defined in terms of the lowpass synthesis filter G_0. The lengths of all the filters is even and is equal to $2K$.

Linear phase [36] is a desirable property for the filters in a perfect reconstruction filter bank used for subband coding. The group delay of a linear phase system is a constant,

$$\tau(\omega) = -\frac{d}{d\omega}\{\mathrm{Arg}(H(e^{j\omega}))\} = \alpha . \tag{3.65}$$

Use of linear phase filters means that the phase distortion from the filtering operation is a simple delay. The only orthogonal perfect reconstruction filter bank with FIR, real coefficient, linear phase filters is the Haar expansion[4], which is described by the low pass synthesis filter

$$g_0[n] = \frac{1}{\sqrt{2}} \qquad n = 0, 1 . \tag{3.66}$$

The Haar expansion is useful for coding bi-level images such as text, but is not useful for subband coding of natural images because the filters used are so short, making the frequency discrimination characteristics of the system poor. In practice, biorthogonal filters tend to give the best performance in compression applications. Biorthogonal systems can achieve perfect reconstruction using FIR, real coefficient linear phase filters. When using subband coding to compress images, the symmetry properties of linear phase filters also lead to effective methods for dealing with edges [49] [50].

General filter banks

Perfect reconstruction systems can be constructed by cascading two channel filter banks in a tree structure. The series expansion nature of perfect reconstruction filter banks makes this intuitively easy to see. In order to concentrate energy in the lowpass bands, subband

3. See Vetterli and Kovačević [55] Appendix 3.A for details.
4. For a proof see Vetterli and Kovačević [55] Section 3.2.4.

coding systems often use octave band spacing of subbands where the lowpass band is the narrowest; this is implemented by iteratively applying the two channel filter bank to the lowpass branch. Figure 3.6 shows the analysis and synthesis filter structure for an iterated filter bank with octave band spacing and recursive decomposition and synthesis of the lowpass band.

Extension of one-dimensional perfect reconstruction filter banks to two dimensions is straightforward when separable filters are used. Filtering, decimation, and interpolation operations are applied to rows and columns separately using one-dimensional filters, decimators, and interpolators. The four resulting two-dimensional subbands are usually labelled LL, LH, HL, and HH, to denote splits into low and high frequency bands in both row and column orientations. Throughout this thesis, the subband labelling convention used is row filtering followed by column filtering; for example the label LH denotes row filtering using the low pass filter followed by column filtering using the high pass filter. Using the test image Lena, Figure 3.7 shows the conventional depiction of the four subbands from a single stage two-dimensional separable subband decomposition. The LL band looks like the original image with each of the two dimensions reduced in size by half. Most of the image energy is concentrated in the LL subband, and there is little energy in the highpass bands, particularly the HH band.

Use of true two-dimensional filters is more complicated than the separable case, and requires two-dimensional up and down sampling operations defined by a 2×2 subsampling matrix, as well as two-dimensional convolution for the filtering operations. To date, use of two-dimensional filters has not shown significant performance gains over the separable application of one-dimensional filters.

Subband coding achieves compression by separating the original signal into different frequency components, and quantizing based on frequency characteristics. It is easy to see that if brickwall filters are used for analysis, components in different subbands are perfectly separated in frequency, but realizable orthogonal filters also produce uncorrelated subbands. Because of the separation into subbands, bit allocation can be made according

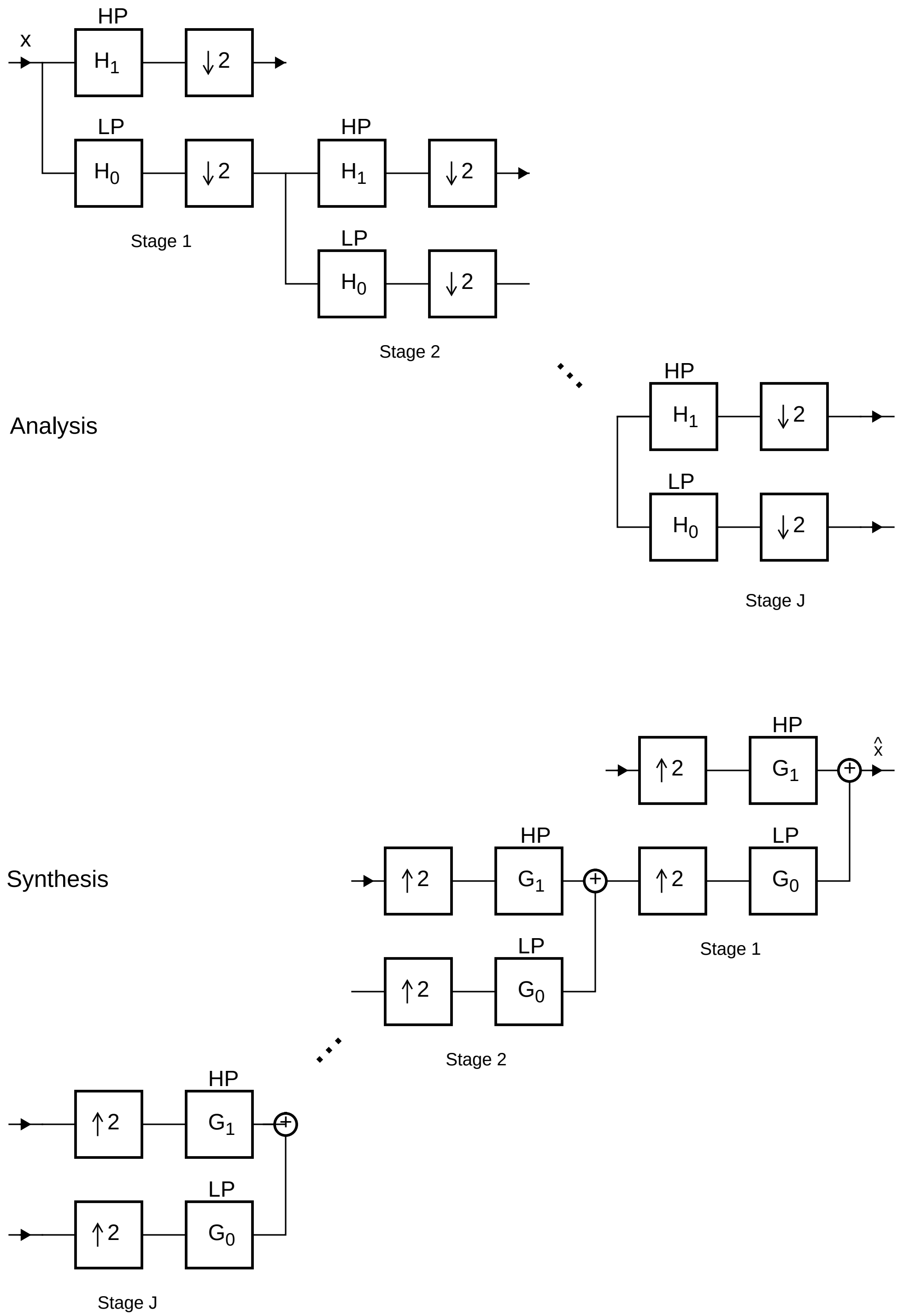


Figure 3.6: Octave band tree structured filter bank

Figure 3.7: Subbands from a separable decomposition

to how the human visual system works in the frequency domain, producing errors in the synthesized image that are perceptually less objectionable. Many natural images tend to have little high frequency content. Decomposing natural images into frequency bands results in energy compaction, because the energy is concentrated in the lowpass band. As in transform coding, energy compaction is used to allocate bits more effectively. For natural images, octave band spacing of subbands produces the most energy compaction for a given number of filtering stages.

Frequency decomposition and subsampling produce a representation of the image made up of components at different resolutions. This multiresolution representation is a powerful concept, leading to efficient quantization algorithms, embedded coding, and linking subband coding to wavelets.

3.6 Wavelets

The connection between wavelets and subband coding is difficult to see at first glance. Wavelet analysis is concerned with continuous time functions represented by continuous time bases, whereas subband coding is concerned with discrete time signals processed by discrete time filters. The design of the filters in a subband coding system is an important factor determining the system's performance. Researchers in the field of wavelet analysis are interested in designing useful wavelets. With the discovery that continuous time wavelets can be generated from discrete time filters and vice versa, the design of wavelets and filter banks have become connected. Filter banks can also be used to efficiently compute the coefficients of a continuous time wavelet expansion using a discrete-time algorithm. Research in both subband coding and wavelet analysis has been active, and advances in one field have driven new research in the other.

The mathematics behind wavelets is quite involved, and any detailed or rigorous discussion is far beyond the scope of this thesis. Textbooks covering wavelets include Vetterli and Kovačević [55], which uses notation and concepts familiar to engineers, and Strang and Nguyen [50]. Useful tutorials on wavelets, written from the perspective of the signal processing community, include the article by Rioul and Vetterli [40], and the short book by Chan [6]. A longer and mathematically more rigorous survey of wavelets is contained in the collection of lectures by Daubechies [16].

3.6.1 Continuous time wavelet analysis

The Continuous Wavelet Transform (CWT) is useful for the analysis of non-stationary continuous time signals. The term compact support is used to describe a function that is non-zero only for a finite duration. The Fourier transform of a signal $x(t)$,

$$X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt, \tag{3.67}$$

is not good at representing signals whose characteristics vary over time, because the basis functions are sinewaves with non-compact support. An alternative is to use the windowed Fourier Transform,

$$X_{WF}(\omega,\tau) = \int_{-\infty}^{\infty} w(t-\tau)x(t)e^{-j\omega t}dt, \tag{3.68}$$

where $w(t)$ is a windowing function and τ represents the shift of the window in time. Because a single window is used, the windowed Fourier transform is restricted to the same resolution for all time and frequency. The Heisenberg inequality,

$$\Delta_t\Delta_\omega \geq \frac{1}{2}, \tag{3.69}$$

where

$$\Delta_t^2 = \frac{\int t^2|x(t)|^2dt}{\int|x(t)|^2dt} \tag{3.70}$$

and

$$\Delta_\omega^2 = \frac{\int \omega^2|X(\omega)|^2d\omega}{\int|X(\omega)|^2d\omega}, \tag{3.71}$$

states that it is not possible to have arbitrarily high resolution in both time and frequency.

The CWT seeks to provide a representation of the signal with good resolution in both time and frequency by using a set of basis functions where some of the basis functions have good time resolution and others have good frequency resolution. The basis functions are constructed by shifting and scaling a single prototype time function known as the mother wavelet. The Continuous Wavelet Transform (CWT) is defined as

$$CWT(a,\tau) = \frac{1}{\sqrt{a}}\int_{-\infty}^{\infty} \psi\left(\frac{t-\tau}{a}\right)x(t)dt, \tag{3.72}$$

where $\psi(t)$ is the mother wavelet, and $x(t)$ is a square integrable function of time. The CWT maps a time function into a function of a and τ. The parameter a represents scale: large scale implies low frequency and small scale implies high frequency. The parameter τ represents shift, the translation of the wavelet function along the time axis. The CWT can be viewed as an inner product with the basis function $\frac{1}{\sqrt{a}}\psi\left(\frac{t-\tau}{a}\right)$, which is a scaled and shifted version of the mother wavelet. Alternatively, it can also be considered as a filtering operation, where the CWT is the output of a bandpass filter with impulse response $\frac{1}{\sqrt{a}}\psi\left(-\frac{t}{a}\right)$, at the instant $\frac{\tau}{a}$. If the Continuous Wavelet Transform is invertible, then

$$x(t) = \frac{1}{c_\psi}\int_{-\infty}^{\infty}\int_{a>0}^{\infty} CWT(a,\tau)\frac{1}{\sqrt{a}}\psi\left(\frac{t-\tau}{a}\right)\frac{1}{a^2}da\,d\tau, \tag{3.73}$$

where c_ψ is a constant that depends only on $\psi(t)$.

When the scale and shift parameters are continuous, the CWT is a highly redundant representation of the signal. Redundancy is reduced by evaluating the CWT on a discrete grid in the time-scale plane. The scale parameter is discretized by setting

$$a = a_0^m, \tag{3.74}$$

and the shift parameter is set to

$$\tau = n\tau_0 a_0^m, \tag{3.75}$$

where a_0 and τ_0 are constants that define the sampling interval, and m and n are integers. The discrete parameter wavelet transform (DPWT) is defined as

$$DPWT(m,n) = \int_{-\infty}^{\infty} \psi_{mn}(t)x(t)dt = \langle\psi_{mn}(t), x(t)\rangle \tag{3.76}$$

where

$$\psi_{mn}(t) = a_0^{-m/2}\psi(a_0^{-m}t - n\tau_0)\,. \tag{3.77}$$

An important particular case of the discrete parameter wavelet transform results from discretization of time and scale on a dyadic sampling grid, where

$$a_0 = 2, \tau_0 = 1\,. \tag{3.78}$$

The discrete parameter wavelet transform on a dyadic sampling grid is commonly but confusingly referred to as the wavelet transform. In Fourier analysis, the term Fourier series describes the expansion of a continuous time function using a discrete set of continuous time basis functions, so the term Continuous Time Wavelet Series (CTWS) is a more descriptive name for the DPWT on a dyadic sampling grid. Writing the definition out explicitly,

$$CTWS(m,n) = \int_{-\infty}^{\infty}\psi_{mn}(t)x(t)dt = \langle\psi_{mn}(t), x(t)\rangle\,, \tag{3.79}$$

where

$$\psi_{mn}(t) = 2^{-m/2}\psi(2^{-m}t - n)\,, \tag{3.80}$$

and m and n are integers. Viewing the CWT as a filtering operation, dyadic sampling means octave scaling of filters.

Reconstruction of the original signal from the discrete parameter wavelet transform representation depends on the sampling rate. Non-unique representations of the original signal with respect to the same mother wavelet are possible if the system is oversampled. If the sampling is sparse then perfect reconstruction is not possible. The theory of frames is applied to define the conditions for perfect reconstruction [15]. When the $\psi_{mn}(t)$ form an orthonormal basis, perfect reconstruction can be defined as

$$x(t) = \sum_m \sum_n \langle x(t), \psi_{mn}(t) \rangle \psi_{mn}(t) \tag{3.81}$$

The orthonormality condition can be relaxed, and perfect reconstruction can be achieved using dual bases $\{\tilde{\psi}_{mn}(t)\}$ and $\{\psi_{mn}(t)\}$,

$$x(t) = \sum_m \sum_n \langle x(t), \tilde{\psi}_{mn}(t) \rangle \psi_{mn}(t) . \tag{3.82}$$

3.6.2 Multiresolution analysis

Multiresolution analysis [33] connects wavelets in the continuous time domain with filter banks in the discrete time domain. Specifically, multiresolution analysis connects the continuous time wavelet series with an octave band subband decomposition using cascaded two channel perfect reconstruction orthogonal filter banks. Multiresolution analysis leads to a method for constructing orthonormal wavelet bases and an algorithm for computing the coefficients of the CTWS.

In a multiresolution analysis, a signal is represented by a coarse approximation and a set of details. The signal is initially decomposed into an approximation component and a detail component using a lowpass/highpass filter pair. The approximation is recursively decomposed into approximation and detail of successively coarser scale. Mathematically, a multiresolution analysis is described as a set of embedded closed subspaces

$$\ldots V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \ldots . \tag{3.83}$$

The set of subspaces is complete, so that

$$\overline{\bigcup_{m \in Z} V_m} = L_2(R) \tag{3.84}$$

and

$$\bigcap_{m \in Z} V_m = \{0\} , \tag{3.85}$$

e.g.

$$\{0\} \rightarrow \ldots V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \ldots \rightarrow L_2(R). \tag{3.86}$$

All other subspaces in the ladder of subspaces are scaled versions of the central space V_0,

$$f(t) \in V_m \Leftrightarrow f(2^m t) \in V_0; \tag{3.87}$$

this property defines the multiresolution aspect of a multiresolution analysis. The central space also exhibits the property of shift invariance for all integer shifts,

$$f(t) \in V_0 \Rightarrow f(t-n) \in V_0 \qquad \text{for all } n \in Z. \tag{3.88}$$

There exists a function of time $\varphi(t)$, known as the scaling function, such that

$$\{\varphi(t-n) \mid n \in Z\} \text{ is an orthonormal basis for } V_0. \tag{3.89}$$

From Equation 3.87, Equation 3.88, and Equation 3.89, a basis for any other subspace in the ladder can be defined,

$$\{2^{m/2}\varphi(2^m t - n) \mid n \in Z\} \text{ is a basis for } V_{-m}. \tag{3.90}$$

Equation 3.83 and Equation 3.90 mean that the scaling function satisfies a two scale equation

$$\varphi(t) = \sqrt{2} \sum_{n=-\infty}^{\infty} g_0[n]\, \varphi(2t-n). \tag{3.91}$$

The sequence $g_0[n]$ can be viewed as a discrete time filter.

The space W_m is defined as the orthogonal complement of V_m in V_{m-1},

$$V_{m-1} = V_m \oplus W_m, \tag{3.92}$$

and

$$W_m \perp W_{m'} \qquad \text{for all } m \neq m' . \tag{3.93}$$

Equation 3.84 and Equation 3.92 mean that the space of square integrable functions can be decomposed into the set of orthogonal subspaces,

$$L_2(R) = \bigoplus_{m \in Z} W_m . \tag{3.94}$$

From Equation 3.87, the set of orthogonal complements also exhibits a scaling property,

$$f(t) \in W_m \Leftrightarrow f(2^m t) \in W_0 . \tag{3.95}$$

It can be shown that

$$\{\psi_{mn}(t) \mid n \in Z\} \text{ is an orthonormal basis for } W_m \tag{3.96}$$

where

$$\psi_{mn}(t) = 2^{-m/2}\psi(2^{-m}t - n) ; \tag{3.97}$$

this together with Equation 3.94, Equation 3.84, and Equation 3.85 means that the $\psi_{mn}(t)$ do in fact form an orthonormal wavelet basis for the space of square integrable functions,

$$\{\psi_{mn}(t) \mid m,n \in Z\} \text{ is an orthonormal basis for } L_2(R) . \tag{3.98}$$

Note that Equation 3.97 is exactly the same as Equation 3.80, connecting multiresolution analysis with the continuous time wavelet series. The wavelet $\psi(t)$ also obeys a two scale equation,

$$\psi(t) = \sqrt{2} \sum_{n=-\infty}^{\infty} g_1[n]\, \varphi(2t - n) , \tag{3.99}$$

and $g_1[n]$ again corresponds to a discrete time filter. The two filters are in fact related: $g_0[n]$ and $g_1[n]$ correspond to the lowpass and highpass synthesis filters in an orthogonal perfect reconstruction filter bank.[5] To reiterate, the lowpass synthesis filter $g_0[n]$ is related to the scaling function $\varphi(t)$ and the subspace V_0, and the highpass synthesis filter $g_1[n]$ is related to the wavelet $\psi(t)$ and the orthogonal complement W_0. Figure 3.8 illustrates the relationship between a few of the different spaces for the sinc wavelet, which corresponds to ideal octave band filters with infinitely steep rolloff.

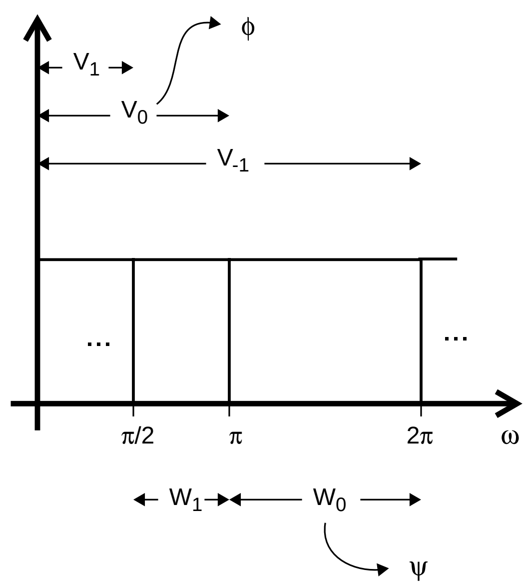


Figure 3.8: Multiresolution analysis using the sinc wavelet

Based on the framework of multiresolution analysis, orthonormal bases of compactly supported wavelets can be constructed from FIR orthogonal perfect reconstruction filter banks [14]. Conversely, perfect reconstruction filter banks can be constructed from wavelets; this has led to improved filters for subband coding. An iterative algorithm is used to generate the continuous time scaling function and mother wavelet from discrete time filters. The support width of the scaling function and wavelet generated using iterated filters is determined by the order of the filters, so that FIR filters lead to wavelet bases with compact support. If the iterative algorithm converges and produces smooth continuous

5. For the complete development of multiresolution analysis as an orthonormal wavelet decomposition, and its relationship to orthogonal filter banks, see Daubechies [16] Chapter 5.

time functions, the filters used are called regular. For orthogonal wavelets, regularity is determined by the frequency domain characteristics of $g_0[n]$. A necessary but not sufficient condition for regularity is that $G_0(e^{j\omega})$, the Fourier transform of $g_0[n]$, have at least one zero at $\omega = \pi$. It is important to note that not all filter banks that satisfy the perfect reconstruction conditions also satisfy the regularity requirement, so not all perfect reconstruction filter banks lead to wavelets. An example of a perfect reconstruction filter bank that does not satisfy the regularity requirement, and therefore cannot be used to construct a wavelet basis, is the 8-tap CQF described by Smith and Barnwell [48].

If the orthonormality constraint is relaxed, dual wavelet bases can be constructed from biorthogonal perfect reconstruction filter banks and vice versa [54]. Biorthogonal wavelets are especially relevant for subband coding applications because biorthogonality permits more freedom in designing the analysis and synthesis filters, and because biorthogonal wavelets are associated with linear phase filters. In the biorthogonal case, the analysis and synthesis filters lead to different wavelets and scaling functions for analysis and synthesis. As in the orthogonal case, the lowpass synthesis filter $g_0[n]$ leads to the synthesis scaling function $\varphi(t)$, and the highpass synthesis filter $g_1[n]$ leads to the synthesis wavelet $\psi(t)$. The analysis scaling function $\tilde{\varphi}(t)$ comes from the analysis lowpass filter $h_0[n]$, and the analysis wavelet $\tilde{\psi}(t)$ is associated with the analysis highpass filter $h_1[n]$. In order for the iterative algorithm to converge, both $h_0[n]$ and $g_0[n]$ must satisfy the regularity condition, so that both $H_0(e^{j\omega})$ and $G_0(e^{j\omega})$ must have at least one zero at $\omega = \pi$.

Multiresolution analysis also results in a discrete time algorithm for computing the coefficients of the continuous time wavelet series using a tree structured octave band filter banks. Starting at the subspace V_0, a continuous time signal $x(t)$ can be written as a linear combination of the basis functions $\{\varphi(t-n)\}$,

$$x(t) = \sum_n c_0[n]\,\varphi(t-n), \tag{3.100}$$

where

$$c_0[n] = \langle x(t), \varphi(t-n) \rangle . \quad (3.101)$$

Equation 3.92 means that the projection of $x(t)$ onto V_0 can be written in terms of the projection onto V_1 and W_1. The projection of $x(t)$ onto the subspace V_1 can be written as

$$x(t) = \sum_n c_1[n] \frac{1}{\sqrt{2}} \varphi(\frac{t}{2} - n) , \quad (3.102)$$

where

$$c_1[n] = \langle x(t), \frac{1}{\sqrt{2}} \varphi(\frac{t}{2} - n) \rangle . \quad (3.103)$$

Expanding the inner product of Equation 3.103, substituting for $x(t)$ using Equation 3.100, and simplifying by applying orthogonality results in

$$c_1[n] = \sum_k g_0[k-2n] c_0[k] . \quad (3.104)$$

Using Equation 3.60 to replace $g_0[n]$ with $h_0[n]$ results in

$$c_1[n] = \sum_k h_0[2n-k] c_0[k] . \quad (3.105)$$

Since these are the equations for the convolution sum with subsampling, $c_1[n]$ can be obtained by filtering $c_0[n]$ with $h_0[n]$, the low pass analysis filter, followed by subsampling by two. The projection of $x(t)$ onto the subspace W_1 can be written as

$$x(t) = \sum_n d_1[n] \frac{1}{\sqrt{2}} \psi(\frac{t}{2} - n) , \quad (3.106)$$

where

$$d_1[n] = \langle x(t), \frac{1}{\sqrt{2}} \psi(\frac{t}{2} - n) \rangle . \quad (3.107)$$

Analogous to the development for V_1, it can be shown that

$$d_1[n] = \sum_k h_1[2n-k]\, c_0[k] \,, \tag{3.108}$$

which means that $d_1[n]$ can be obtained by filtering $c_0[n]$ with $h_1[n]$, the high pass analysis filter, followed by subsampling by two. Together, Equation 3.92, Equation 3.105 and Equation 3.108 mean that the projection of $x(t)$ onto any V_m and its complement W_m, $c_m[n]$ and $d_m[n]$ respectively, can be obtained from the projection of $x(t)$ onto V_{m-1} by discrete time filtering and subsampling. The coefficients of the continuous time wavelet series can be computed by iteratively applying two channel perfect reconstruction filter banks in the topology shown in Figure 3.9. But the filter tree in Figure 3.9 is the same as the analysis filter tree in Figure 3.6, establishing the connection between the CTWS and perfect reconstruction filter banks.

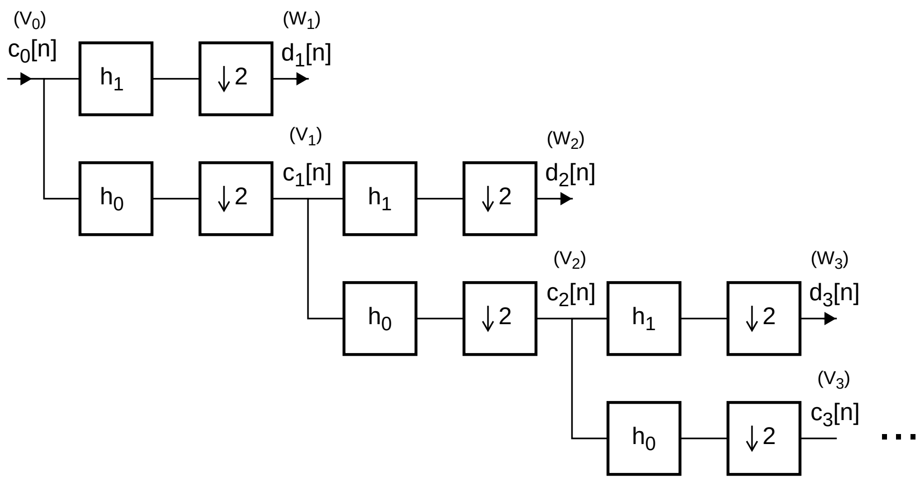


Figure 3.9: Calculation of CTWS coefficients

3.7 Image compression

Wavelet analysis is a powerful tool, and mathematically the connection between wavelets and filter banks is well understood. The use of filter banks in subband coders leads to the application of wavelets to the image compression problem. The insight from the wavelet point of view has led to improvements in filter design, as well as new image compression methods such as wavelet packets.

3.7.1 The wavelet transform

The close relationship between wavelet analysis and subband coding has made the term “wavelet transform” synonymous with a tree structured subband decomposition with octave band spacing where the lowpass band is the narrowest. This use of the term wavelet transform is common in the literature, but is somewhat confusing because it does not explicitly make clear that a discrete time signal is analyzed using discrete time basis functions. Antonini et al. [1] investigate the use of the wavelet transform to compress images. Three sets of biorthogonal filters are tested: a set of filters where the lowpass and highpass filters have dissimilar lengths (9-3)[6], a set of filters where the lowpass and highpass filters have similar lengths (9-7), and a set of filters where the analysis and synthesis wavelets are similar (5-7). The wavelet coefficients are quantized using vector quantization. Different codebooks for each subband are used, and the codebooks are generated using the Linde-Buzo-Gray algorithm with an MSE criterion. Bit allocation for each subband is optimized based on the human visual system. In terms of PSNR, the results using the wavelet transform are comparable to the performance of state of the art coders at the time, while using shorter filters. The best results are achieved with the 9-7 filters, and the authors claim that the perceptual performance with this filter set at low bit rates is better than the state of the art at the time. The 9-7 biorthogonal filter set is very commonly used in the literature, since it provides a good overall partition even with other coding techniques.

An important question when using the wavelet transform for image compression is how the characteristics of the wavelets associated with the filter bank are related to the

6. In the literature, the filters in a biorthogonal two channel filter bank are often referred to by the length of the analysis filters e.g. 9-3, where 9 is the length of the lowpass filter and 3 is the length of the highpass filter.

performance of the image compression system. If a wavelet $\psi(t)$ has N vanishing moments, then

$$\int_{-\infty}^{\infty} t^n \psi(t) dt = 0, \qquad n = 0, \ldots, N-1. \tag{3.109}$$

Regularity is a measure of the smoothness of a wavelet. Antonini et al. propose that a filter bank where the associated analysis wavelet has a large number of vanishing moments, and the synthesis wavelet has a high degree of regularity, works well for image compression. However, a recent study by da Silva and Ghanbari [12] suggests that regularity and number of vanishing moments have little correlation with image compression performance. They investigate subjective and objective compressed image quality using a large number of regular biorthogonal filters. Image quality is measured with both a subjective grade and PSNR. Image quality is studied at bit rates down to 0.5 bits per pixel using a 3 stage separable wavelet transform decomposition. The lowpass band is coded with PCM at 8 bits per pixel, and all other coefficients are coded using a multistage product code lattice based vector quantizer. The only characteristic that shows any relationship with image quality is the shape of the synthesis wavelet, where a more concentrated wavelet is associated with better performance.

3.7.2 Zerotree coding

Zerotree coding is an image compression scheme that exploits the characteristics of the subband decomposition to quantize and code subband coefficients efficiently. The advantages of zerotree coding include good rate-distortion performance, computational efficiency, and embedded coding. Zerotree coding is especially effective at maintaining good perceptual quality at low bit rates. Scalar quantization followed by entropy coding is at the heart of zerotree coding, but it is the clever way that the quantization and coding is done that makes zerotree coding so effective. The concept of zerotree coding was first introduced by Shapiro [46] as the Embedded Zerotree Wavelet algorithm (EZW), and has been extended by Said and Pearlman [42] into a technique that they call Set Partitioning In

Hierarchical Trees (SPIHT). Because Said and Pearlman's codec works very well in terms of both speed and performance, and because it is freely available in source code and executable form on the Internet[7], it serves as a useful standard high performance wavelet coder for investigating image compression problems.

One of the key ideas in zerotree coding is ordered bit plane transmission of subband coefficients. Effectively, coefficients are ordered by magnitude, then the bits in the same bit position of all coefficients are transmitted, high order bits first. Ordered bit plane transmission produces an embedded code and effectively allocates more bits to large magnitude coefficients. Because of energy compaction by the subband decomposition, there are few ones to be coded in the high order bit positions. The converse is that there are many zeros in the high order bit positions to be represented. Efficient representation of zeros leads to the second key idea in zerotree coding: the magnitude of coefficients in the same spatial location are related across scales. This relationship depends on the multiresolution aspect of the subband decomposition. The result is that zerotree coding represents a group of coefficients in the same spatial location across scales that are less than a threshold value using a single symbol; this group of coefficients is called a zerotree. For further efficiency, the stream of symbols representing significant bits and zerotrees is entropy coded using adaptive arithmetic coding.

In order to perform well, zerotree coding relies on some assumptions that are generally valid for natural images, but these assumptions produce some problems when coding ultrasound images. The wavelet transform is a reasonable choice of frequency decomposition topology for natural images, and allocating bits by coefficient magnitude works well when the contrast level is quite uniform. When used to compress ultrasound images, zerotree coding produces some distinctive artifacts. Speckle is blurred at lower bit rates, suggesting that a different decomposition topology is better suited to speckly images. Because contrast is important in an ultrasound image, bit allocation by coefficient magni-

---

7. See the SPIHT web site at `ipl.rpi.edu/SPIHT`.

tude is not the best choice, and artifacts caused by this bit allocation scheme are particularly visible in areas of low contrast detail.

3.7.3 Wavelet packets

The concept of wavelet packets extends the octave band tree structured filter bank to include all binary frequency decompositions, so that the best decomposition topology can be chosen to suit the characteristics of an individual image. In continuous time, wavelet packets lead to the construction of wavelet bases for $L_2(R)$ for the discrete parameter wavelet transform that are not constrained by the dyadic sampling requirement of multiresolution analysis. Wavelet packets were first introduced by Coifman et al. [9]

In a tree structured filter bank, iterative decomposition of the low pass branch is only one possible topology for the tree. Starting from a two channel orthogonal perfect reconstruction filter bank, the choice can be made to further decompose the highpass branch, the lowpass branch, neither branch, or both branches. At each stage in the tree, the same decision can be made for each leaf. Since the filtering operation can be viewed as a projection onto a set of basis functions, the tree structure represents a choice of basis. For a one-dimensional signal, S_J, the number of possible bases from a tree of depth J, is described by the recursive equation

$$S_J = S_{J-1}^2 + 1, \tag{3.110}$$

where the undecomposed signal is represented by

$$S_0 = 1. \tag{3.111}$$

In Equation 3.110, the S_{J-1}^2 term includes bases from the lowpass and highpass branches, and the additional basis is the undecomposed case. Figure 3.10 graphically illustrates a one-dimensional wavelet packet decomposition of maximum depth two, and lists the five possible representations.

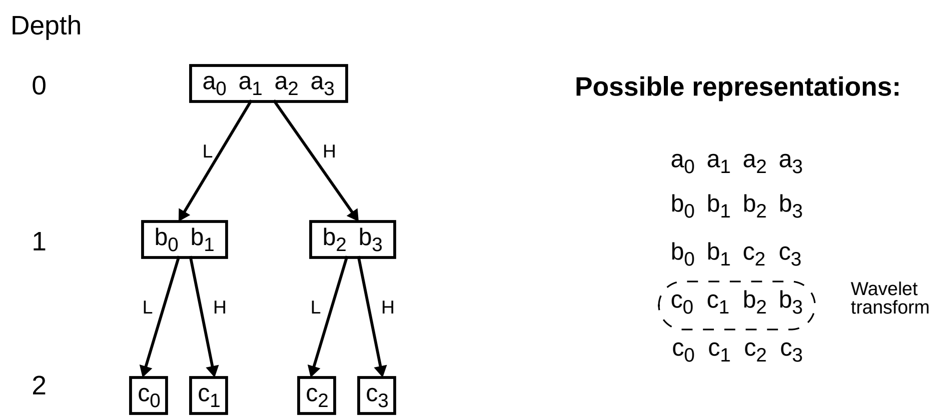


Figure 3.10: Wavelet packet decomposition, maximum depth = 2

The best basis needs to be chosen from the set of possible bases, so a criterion for choosing the best basis must also be defined. For compression, Coifman et al. propose minimum entropy as the criterion for determining the optimum tree structured filter bank. Ramchandran and Vetterli [39] connect the quantization and transformation components of lossy compression systems by choosing optimum rate-distortion performance as the criterion for basis selection, and describe a fast tree search based algorithm for determining the best basis. The Ramchandran and Vetterli algorithm is discussed in more detail in Section 5.3.

3.7.4 Comparing image codecs

To give some quantitative perspective on the different image compression methods that have been discussed, Table 3.5 lists the PSNR at 1.0 bpp and 0.5 bpp obtained by coding the test image Barb[8] (shown in Figure 3.11) using JPEG, EZW, wavelet packets, and SPIHT. Because a JPEG codec[9] and the SPIHT codec are freely available on the Internet, the values in Table 3.5 for JPEG and SPIHT are obtained by experiment, whereas the val-

8. The Gersho version of Barb is used to obtain the experimental results for JPEG and SPIHT.
9. The JPEG codec is obtained from the Independent JPEG Group, www.ijg.org.

Table 3.5: Barb compressed at 1.0 bpp and 0.5 bpp

Method	R (bpp)	PSNR (dB)
JPEG	1.0	33.3
EZW	1.0	35.1
Wavelet packets	1.0	37.1
SPIHT	1.0	36.4
JPEG	0.5	27.8
EZW	0.5	30.5
Wavelet packets	0.5	30.3
SPIHT	0.5	31.4

ues for EZW and wavelet packets are the published values in Shapiro's [46] and Ramchandran and Vetterli's [39] papers, respectively. Of the four coding methods, JPEG produces the worst PSNR results by a considerable margin at both 1.0 bpp and 0.5 bpp. Wavelet packets performs the best at 1.0 bpp, but at 0.5 bpp both zerotree codecs perform better. Overall, SPIHT appears to work the best, producing the best PSNR at 0.5 bpp and the second best PSNR at 1.0 bpp.

Figure 3.11: Barb

Even this simple comparison of coding methods on a single well accepted test image is not as straightforward as it looks, underlining the care that must be taken when comparing compression results in the literature. The rate results for wavelet packets reported by Ramchandran and Vetterli are zeroth-order entropy estimates, and are not actual coded rates. As will be shown in Section 6.3, the zeroth-order entropy is an optimistic estimate of rate, especially when coding small subbands. Even when comparing actual coded rates, it is difficult to make an apples-to-apples comparison of coding algorithms. All three subband algorithms use different filters: Shapiro uses 9-tap QMFs, Ramchandran and Vetterli use the Daubechies 8-tap orthogonal filters, and Said and Pearlman use the Antonini 9-7 biorthogonal filters. It is quite likely that the EZW results can be improved simply by using the same filters as SPIHT, so the advantage of SPIHT over EZW is not as large as the PSNR values in Table 3.5 seem to indicate.

Chapter 4

Compression and Ultrasound Images

4.1 Recent research

In a recent article, Erickson et. al. [18] review the compression of medical images using the wavelet transform. They discuss compression of images from a variety of medical imaging modalities, including computed tomography (CT), magnetic resonance imaging (MRI), chest radiography, and ultrasound. JPEG is compared against wavelet transform methods such as SPIHT in their literature review and their own experiments. The conclusion, based primarily on subjective quality assessment, is that in general wavelet transform techniques are superior to JPEG, due to the blocking artifacts produced by JPEG. Their comments on compression of ultrasound images highlight the speckle texture as a difficult feature to compress, and they suggest that relative to other modalities ultrasound produces images that are less tolerant to compression. Using SPIHT, they report that typically ultrasound images cannot be compressed at rates greater than 20:1 without degrading the image beyond acceptability in terms of aesthetic appearance and utility for medical interpretation.

Krasner et. al. [31] apply vector quantization to compress ultrasound liver images, and discuss the effect of compression on image quality. Their work is interesting because in addition to characterizing compressed image quality using a conventional squared error metric, they also study the effect of compression on image statistics that are useful for tissue characterization. Images are compressed using two methods: pruned tree structured vector quantization (PTSVQ), and PTSVQ with splitting. PTSVQ with splitting involves dividing coefficients into high and low order bits. High order bits are compressed losslessly using Lempel-Ziv coding, while low order bits are compressed using PTSVQ. They find that PTSVQ with splitting results in less degradation in terms of mean squared error, but more degradation in terms of statistical tissue characterization metrics, which shows that mean squared error can be misleading. A disadvantage of this study is that it is restricted to relatively low compression rates, generally above 1.0 bpp.

Cabral, Linker and Kim [5] compress pre-scan-converted images with JPEG and a wavelet transform codec, comparing the results against the compression of scan-converted ultrasound images, which is the conventional case. (See Section 2.4 for a brief description of scan conversion.) The wavelet transform codec uses a depth three decomposition, EZW bit allocation and quantization, followed by arithmetic coding. Experiments using the wavelet codec are done with a variety of filters, including orthogonal Daubechies 4, 8, and 12 tap filters, and biorthogonal 9-3 and Antonini 9-7 filters. The Antonini 9-7 filters show the best performance. Their results show that the wavelet transform EZW codec is superior to JPEG, and that compressing pre-scan-converted images produces better results than compressing scan-converted images. At 0.5 bpp and averaged over nine images, the improvement in PSNR for JPEG of pre-scan converted over scan-converted images is 2.5 dB, improving from an average of 31.3 dB to 33.8 dB. For the wavelet codec, the average improvement in PSNR is 0.7 dB, from 34.3 dB to 35.0 dB, by compressing pre-scan-converted images. In equivalent comparisons (scan-converted versus scan-converted results, pre-scan-converted versus pre-scan-converted results), the wavelet transform EZW codec outperforms JPEG in both PSNR and subjective quality. In their article they show the recovered image of a pre-scan-converted image compressed at 0.37 bpp, and report that in terms of subjective quality the EZW compressed image is almost identical to the uncompressed image. They propose that the improved performance from compressing pre-scan-converted images is due to the increased correlation between raster image pixels in scan-converted images as the distance from the transducer increases. The drawback to compressing pre-scan-converted images is that the image must be scan converted in addition to being decompressed before display. The scan-conversion parameters are determined by the transducer and acquisition equipment, so portability of the image is restricted unless these parameters are included with the image.

Luo, Li and Chen [32] identify speckle as a problem in the compression of ultrasound images, and try a novel solution, however their results do not appear promising. Subband coding is at the heart of their technique. The original ultrasound image is decomposed into subbands, certain high pass subbands with speckle energy content are set to zero, then the subbands are synthesized, producing what they call a structure image. They generate what they call a speckle image by subtracting the structure image from the original image. The structure image is encoded using SPIHT, but the entire speckle image is not explicitly encoded. Instead, a small rectangular portion of the speckle image is losslessly encoded and sent along with the SPIHT encoded structure image. At the receiver, the decoded speckle patch is used as input to a texture model to synthesize a speckle image, which is added to the decoded structure image to generate the final reconstructed estimate. They report results only for a single image, and in terms of PSNR, their technique performs poorly when compared against simply coding the entire original image using SPIHT. At 0.547 bpp, the PSNR of the SPIHT coded image is 29.6 dB, but the PSNR of the synthesized speckle plus structure image is only 21.5 dB. This poor PSNR result is expected, since there is only random correspondence between individual speckle spots in the synthesized speckle image and the original image. The authors argue that the appearance of the image reconstructed using their method is superior to SPIHT, but it is difficult to draw this conclusion from the pictures included in their article.

de Solà Fàbregas and Tri [17] focus on the shape of the ultrasound-only region in applying a shape adaptive DCT to compress sector scanned ultrasound images. The transformation still operates on an 8×8 block of pixels, but a region of interest mask separates pixels in a block into foreground and background pixels, and a separable transformation is applied to foreground pixels only. Like JPEG, the transformed coefficients of a block are then zig-zag scanned into a one-dimensional sequence, quantized, then run-length and entropy coded. Results are reported for a single echocardiographic image, which for reference is also coded using JPEG to a PSNR of 42 dB at 0.43 bpp. Using the shape adaptive DCT, experiments with a selection of different quantization tables result in an improved PSNR of 42.6 dB at a lower rate of 0.32 bpp. Drawbacks for region of interest type coding schemes which the authors do not address are the extra complexity required in the encoder to define the region of interest, and the need to specify the region of interest to the decoder as side information. This article also highlights a problem with reported medical image compression results. Because there are many different modalities, and no standard test images even within the same modality, reported compression results are difficult to compare. The reported PSNRs at the reported bit rates in this article seem unusually high relative to results from the other reviewed articles. It seems intuitive that it is a good idea to code a region of interest rather than the entire image when significant portions of the image are background, but without knowing any more about the test images used, it is difficult to make any further assessments of the results from this article.

It is not easy to summarize the current state of the art for the compression of ultrasound images. Compared with other modalities, the literature on compression specifically of ultrasound is limited. In the articles reviewed, the lack of common test images makes it difficult to compare results in different articles. Nevertheless, three of the five articles discussed (Erickson et. al., Cabral et. al., Luo et. al.) apply zerotree coding to compress ultrasound images, and two of these three articles (Erickson and Cabral) compare zerotree coding against JPEG and conclude that zerotree coding is superior. Based on the reviewed articles, it appears that zerotree coding is currently the leading technique for the lossy compression of medical ultrasound images.

4.2 Subband decomposition and image content

To connect the variety of background material that has been presented on ultrasound images, image compression, subband coding, and wavelets, a single stage subband decomposition is applied to a normal image and an ultrasound-only section of a medical ultrasound image that can be considered typical. The representative normal image is a 256×256 pixel version of the test image Lena[1]. The ultrasound-only image is labelled Ub, and is also 256×256. Ub is cropped from a sector scanned image of a liver[2]. The two images are shown in Figure 4.1. Orthogonal filters, the Daubechies filters of length eight, are used in order to compare the energy in the different subbands precisely.

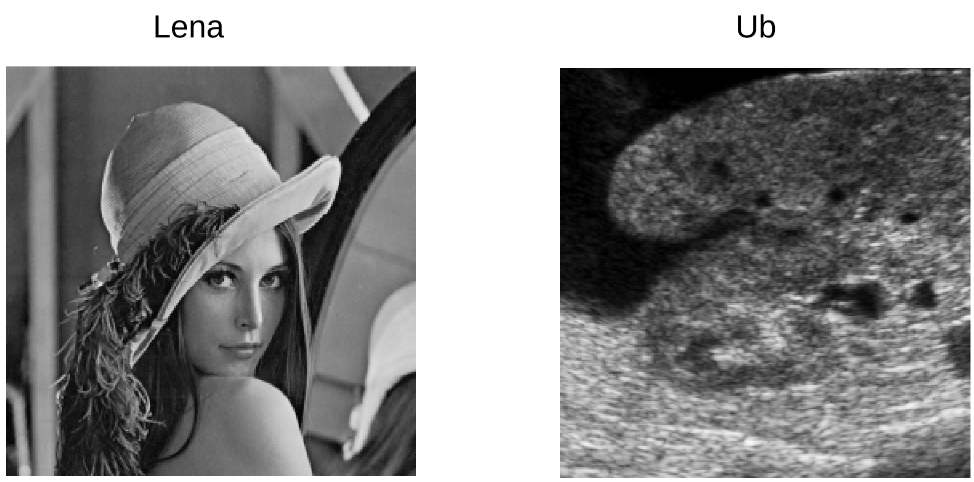


Figure 4.1: Lena and Ub

The subband images for the two images after single stage subband decomposition are shown in Figure 4.2. For both images, over 98% of the energy is concentrated in the LL band. The distribution of the remaining 2% of the energy in the high pass bands is reported in Table 4.1, which shows that the high pass energy distribution of Ub is much less uniform than that of Lena. Edges and details from the original image appear as features with reduced energy in all three of the high pass subbands of Lena. Features such as

1. This version of Lena is obtained from the Waterloo Bragzone at links.uwaterloo.ca/greyset1.base.html.
2. The original image is obtained from the ALI Technologies website, www.alitech.com. ALI is a manufacturer of medical PACS.

Table 4.1: High pass energy distribution

Image	Subband	% of HP energy
Lena	LH	22%
Lena	HL	64%
Lena	HH	14%
Ub	LH	88%
Ub	HL	10%
Ub	HH	2%

the boundary between Lena's shoulder and the background, the outlines of her eyes, nose and mouth, the edges of her hat, and the feathers from her hat are all discernible in the high pass subbands. There are more vertical than horizontal features in Lena, which is nor-

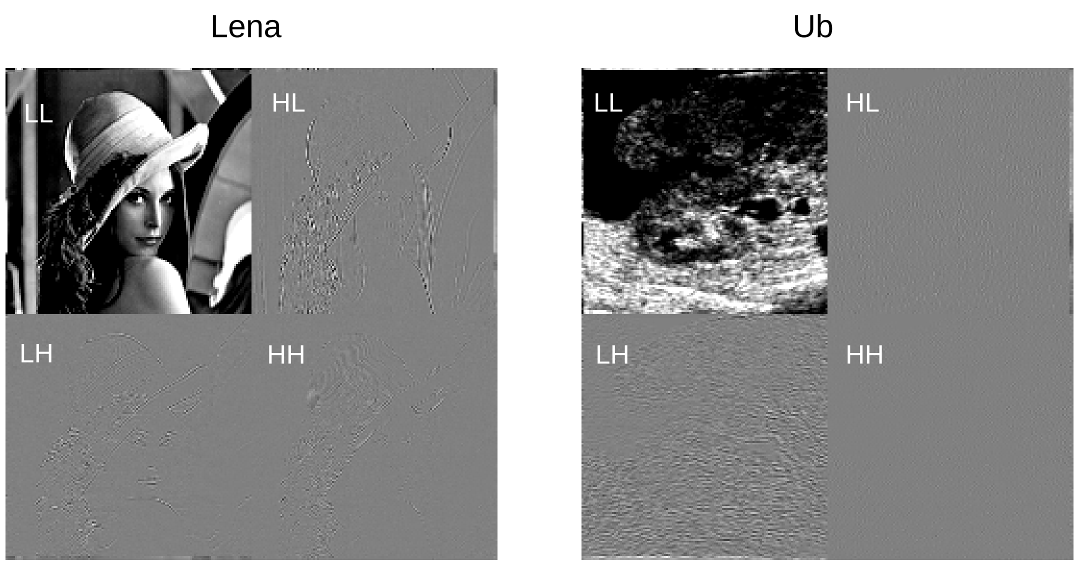


Figure 4.2: Subband images: Lena and Ub

mal for an image of a person, so there is more energy in the HL band than the other two high pass bands. In contrast, there are few edges in the ultrasound-only image, which is dominated by speckle. After subband decomposition of the ultrasound image, the speckle is concentrated in the LH band, and no features are visible in the HL or HH bands. For the ultrasound image, almost all the high pass energy is in the LH band. The concentration of high pass energy is explained by the shape of a speckle spot. Speckle spots in ultrasound images are generally enlongated, due to the different transducer resolution in the lateral and axial directions. Often the image is oriented such that the direction of insonification is vertical, so that the speckle spot is elongated in the horizontal direction.

This simple experiment comparing the single stage subband decomposition of Lena and Ub highlights an important issue in compressing ultrasound images. Ultrasound images are quite different from normal images, and their features are dominated by speckle. An efficient codec for normal images may not work as well for ultrasound images, but improved compression performance may be obtained by applying techniques that adapt to the unusual characteristics of ultrasound images.

Chapter 5

Space-frequency Segmentation

5.1 Introduction

Space-frequency segmentation is an adaptive lossy compression technique that represents an image as a set of quantized subimages. The algorithm finds the optimal set of subimages by choosing the optimal decomposition topology from a finite set of possible decomposition topologies, and finds the optimal bit allocation by choosing the best quantizer for each subimage from a finite set of quantizers, all chosen according to a rate-distortion criterion.

The possible decomposition topologies are determined by the type of decompositions allowed. In space-frequency segmentation, decomposition in either space or frequency produces four subimages. For space-frequency segmentation, decomposition in space means dividing an image into four quadrants, and decomposition in frequency means filtering an image into four subbands using a two-dimensional separable perfect reconstruction filter bank. Each subimage can also be decomposed in space or frequency, up to some maximum decomposition depth, so that the optimal set of subimages is generated by a particular subtree chosen from a tree of space and frequency decompositions. As discussed in Section 3.5, filtering with a perfect reconstruction filter bank can be viewed as a projection onto a set of basis functions, so the space-frequency tree defines a set of possible bases, and the optimal subtree defines a specific basis. It will be shown that space-frequency segmentation searches a larger set of possible bases than the wavelet packet double-tree algorithm [26].

5.2 Rate-distortion theory

Space-frequency segmentation determines the best basis and choice of quantizers in a rate-distortion sense, which means that for a given target rate, the algorithm chooses the representation that will result in the lowest distortion. The optimal representation corresponds to the operating point on the convex hull of the operational distortion-rate function that has a rate that is closest to but does not exceed the target rate. In order to understand what this means, a brief discussion of rate-distortion theory is required.

Rate-distortion theory [45] [3] [25] applies the mathematical rigor of information theory to the problem of data compression. It relates the information produced by a source to the fidelity with which that information can be recovered by the user. A limitation of rate-distortion theory is the difficulty in defining measures of information and fidelity for real world sources that are both practically relevant and mathematically tractable. Despite its theoretical nature, rate-distortion theory is useful to the design of data compression systems because it provides a performance bound for the system.

In a general communication system, a source conveys information to a user. The rate-distortion function, $R(D)$, defines the relationship between the user and the source by stating that the user can recover information with distortion D if and only if information is received at a rate that exceeds $R(D)$. A typical rate-distortion function is shown in Figure 5.1. An important property of the rate-distortion function is that it is a continuous, monotonic decreasing convex function for $0 \le D < D_{max}$, and is equal to zero for $D > D_{max}$.

It is often useful to define the source-user relationship in the inverse manner in terms of the distortion-rate function, $D(R)$. For a rate constraint R, the minimum achievable distortion is given by $D(R)$. In practice, the distortion-rate function and rate-distortion function are often used interchangeably.

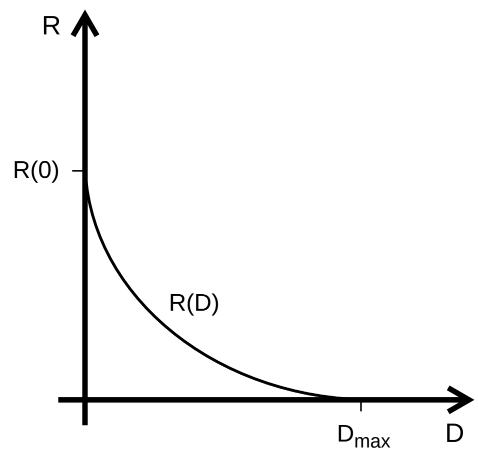


Figure 5.1: Typical rate-distortion function

When a source is quantized with a finite set of quantizers, the distortion-rate constellation of all achievable distortion-rate pairs does not yield a continuous distortion-rate function. Instead, a staircase line referred to as the operational distortion-rate function joins the set of points that minimize distortion at a given achievable rate. The convex hull of the operational distortion-rate function consists of straight lines connecting points on the operational distortion-rate curve such that no point on the operational distortion-rate curve is below the convex hull. A distortion-rate constellation, the operational distortion-rate curve, and its convex hull are shown in Figure 5.2.

5.3 Optimal wavelet packet bases

Single-tree wavelet packets can be viewed as an adaptive binary subband decomposition, where the decomposition topology for a particular source is chosen with respect to some optimality criterion. Ramchandran and Vetterli [39] solve the problem of choosing the best wavelet packet basis and best quantizer for each subband in a distortion-rate sense. The Ramchandran and Vetterli algorithm extends the Generalized BFOS[1] algorithm of

1. The acronym BFOS stands for Breiman, Friedman, Olshen and Stone.

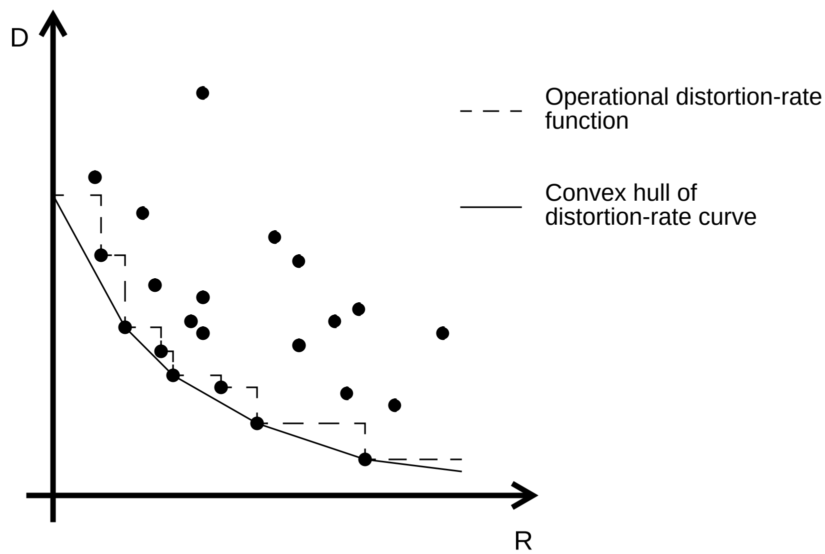


Figure 5.2: Operational distortion-rate

Chou, Lookabough and Gray [7], which finds the distortion-rate operating points of a tree structured problem, by removing the monotonicity constraints that Generalized BFOS places on the tree. Ramchandran and Vetterli also extend the optimal bit allocation algorithm of Shoham and Gersho [47], which finds the operating point on the convex hull of the distortion-rate function in terms of the slope at that point, by structuring the problem using a tree and searching for the optimal slope as a maximum of a convex function. The Ramchandran and Vetterli technique is very powerful, and can be applied to the general distortion-rate optimization of a tree structured problem with an arbitrary set of quantizers. It will be used to find the best space-frequency partition, and to choose the best quantizer for each group of space-frequency coefficients.

A single-tree wavelet packet decomposition is represented by a tree, which defines the possible topologies of a tree structured filter bank. Only frequency decomposition, filtering by a perfect reconstruction filter bank, is allowed in a single-tree decomposition. Each node in a single-tree corresponds to a subband, and each branch in the tree corresponds to the decomposition into a subband, so that the children of each node correspond to the subbands obtained by filtering the parent subband. At each node, there is a decision as to whether or not to decompose the corresponding subband, subject to some maximum decomposition depth constraint. When there are no further decompositions below a node, the corresponding subband is quantized, so there is also a decision of the choice of quantizer from a finite set of possible quantizers. The collection of decisions about whether or not to decompose in frequency defines an optimal subtree, and the collection of decisions about the quantizers at each leaf node in the subtree defines the optimal quantizer set for that subtree. The decomposition and quantizer choices are made according to a rate-distortion criterion; together, they define an optimal decomposition and bit allocation. A closer look at Ramchandran and Vetterli's development of a solution follows, showing how the search for an optimal decomposition and bit allocation is converted to an expression that is easily solved.

Some definitions are required. Let T represent the maximum depth wavelet packet tree, $S \leq T$ represent a pruned subtree of T, and $\tilde{S}$ represent the set of terminal nodes of the subtree S. Corresponding to each node n in the tree is a choice of basis and a set of coefficients. Each $\tilde{S}$ defines a unique subband decomposition, and together with the set of quantizers used for each node, it specifies the codec. Let $q_a(n)$ denote the set of admissible quantizers for node n, and $Q_a(S)$ denote the set of admissible quantizers for the leaf nodes of subtree S. Assume that the quantized coefficients at each node will be coded independently using an arbitrary entropy coding scheme. $D_q(n)$ and $R_q(n)$ represent the distortion and rate for coding the coefficients at node n with quantizer $q \in q_a(n)$. Assume that squared error is used as the distortion measure, and entropy is used as the rate measure. The distortion $D_Q(S)$ and rate $R_Q(S)$ for coding subtree S with quantizer set $Q \in Q_a(S)$ is

$$D_Q(S) = \sum_{n \in \tilde{S}} D_q(n) \qquad (5.1)$$

and

$$R_Q(S) = \sum_{n \in \tilde{S}} R_q(n). \tag{5.2}$$

Ramchandran and Vetterli find a solution by converting a constrained problem to an unconstrained problem in terms of the Lagrange multiplier λ. The parameter λ is also the slope of the convex hull of the distortion-rate function, and λ^*, the value of the Lagrange multiplier that solves the unconstrained problem, corresponds to the singular point on the convex hull that is closest to but does not exceed the rate constraint. Solving the constrained problem,

$$D_{Q^*}(S^*) = \min_{S \le T}\left[\min_{Q \in Q_A(S)} D_Q(S)\right] \tag{5.3}$$

such that

$$R_{Q^*}(S^*) \le R_{\text{budget}}, \tag{5.4}$$

finds the best basis and best quantizer set that minimizes the distortion for a given rate constraint. Define $J_Q(S, \lambda)$ as the Lagrangian cost function corresponding to the Lagrange multiplier λ,

$$J_Q(S, \lambda) = D_Q(S) + \lambda R_Q(S). \tag{5.5}$$

Solving the unconstrained problem,

$$J_{Q^*}(S^*, \lambda) = \min_{S \le T}\left[\min_{Q \in Q_A(S)} [D_Q(S) + \lambda R_Q(S)]\right], \tag{5.6}$$

is equivalent to solving the constrained problem when

$$R_{\text{budget}} = R_{Q^*}(S^*). \tag{5.7}$$

The goal is to find a way to efficiently compute the solution to the unconstrained problem. Let x represent the distortion-rate operating point associated with a particular choice of subtree and quantizer set. The Lagrangian cost can be written in “flattened” form as

$$J(\lambda) = D(x) + \lambda R(x). \tag{5.8}$$

Using $J(\lambda)$, define the biased Lagrangian cost functional $W(\lambda)$,

$$W(\lambda) = \min_{x} [D(x) + \lambda R(x)] - \lambda R_{\text{budget}}. \tag{5.9}$$

Ramchandran and Vetterli prove that $W(\lambda)$ is a convex function. Let λ^* denote the value of λ that maximizes $W(\lambda)$, with corresponding distortion-rate operating point $x(\lambda^*)$. Ramchandran and Vetterli show that λ^* is the convex hull slope and $x(\lambda^*)$ is the distortion-rate operating point that solves the unconstrained problem with target rate R_{budget}. The problem is now to find the value of λ that maximizes $W(\lambda)$, and since $W(\lambda)$ is a convex function, the solution is well behaved. The solution to the unconstrained problem can be written in terms of the biased Lagrangian cost in “unflattened” form as

$$W(\lambda^*) = \max_{\lambda \geq 0}\left(\left[\min_{S \leq T} \left\{ \sum_{n \in \tilde{S}} \min_{q \in q_a(n)} [D_q(n) + \lambda R_q(n)] \right\} \right] - \lambda R_{\text{budget}} \right). \tag{5.10}$$

Equation 5.10 summarizes the Ramchandran and Vetterli algorithm, and consists of three nested optimizations. The innermost minimization chooses the quantizer that produces the lowest Lagrangian cost for each node in the tree, for a given value of the Lagrange multiplier λ. The middle minimization finds the minimum cost subtree, again for a given value of λ, using a fast tree pruning algorithm. At a tree node, if the cost of the children is less than the cost of the parent (current) node, the winner is the children, otherwise the winner is the parent, and the node inherits the cost of the winner. The orthonormality property of wavelet packet bases allows direct comparison between the cost of parent and children. Starting from the bottom of the tree, this tree pruning rule is applied recursively to each node in the tree, so that the winning costs “bubble up” in the tree level by level, until the root node is reached with the cost of the optimal tree. The outermost maximization finds the value of λ that meets the rate constraint by maximizing the biased

Lagrangian cost function. The search for λ^*, the slope of the convex hull that defines the operating point, is iterative. The algorithm starts with an upper and lower limit for λ^*, and refines this range using a bisection method until it converges.

The Ramchandran and Vetterli algorithm finds the best basis and quantizer set for a tree structured problem with a set of arbitrary quantizers. The algorithm development relies on the use of the biased Lagrangian cost functional $W(\lambda)$ (Equation 5.9 and Equation 5.10). Ramchandran and Vetterli use the parameter λ to find a solution, and search rate-distortion operating points that are singular points on the convex hull, using an efficient tree pruning scheme. Because $W(\lambda)$ is a convex function, Ramchandran and Vetterli use a fast bisection algorithm to find the operating point.

5.4 Space-frequency segmentation

In the Ramchandran and Vetterli algorithm to find the optimal wavelet packet basis, the middle optimization of Equation 5.10 finds the minimum cost subtree for the current value of λ. The tree topology corresponds to a choice of basis, chosen from a set of possible bases that is determined by the type of decomposition and type of structure allowed in the tree. By replacing the wavelet packet single-tree with a different tree in the best subtree search, the Ramchandran and Vetterli algorithm can be directly applied to find the distortion-rate optimal basis and quantizers for a different set of possible bases. Space-frequency segmentation [27] uses a tree that is symmetric in space and frequency partitions to produce a rich set of possible bases that can be searched efficiently. The space-frequency segmentation algorithm is best understood by examining its one-dimensional counterpart, time-frequency segmentation. A variation on space-frequency segmentation, the “block” algorithm, is faster than the general algorithm but only searches a subset of the possible space-frequency bases. If the bases excluded by the “block” space-frequency algorithm are not useful for image compression applications, this algorithm can be used instead of the general algorithm to obtain a close to optimal solution.

5.4.1 The one-dimensional case: time-frequency segmentation

One approach to finding a useful alternative tree topology is to grow wavelet packet trees over binary time segmentations of the original one-dimensional signal. This is known as the double-tree algorithm [26]. A wavelet packet single-tree is grown over the entire signal, then the original signal is split in half and a single-tree is grown over each half. The depth of the two trees grown over the binary time segments is one less than the tree over the undivided signal. The process can be repeated over quarters, eighths, et cetera until the desired level of segmentation is reached, or the signal can no longer be split into binary segments. Each single-tree for a segment is pruned using the wavelet packet tree pruning algorithm, and the winning cost is stored in a binary cost tree. Pruning this cost tree produces the best double-tree basis for the current cost function. The number of possible bases for a depth J decomposition is described by the recursive equation

$$D_J = D_{J-1}^2 + S_J - 1, \tag{5.11}$$

where

$$D_0 = 1, \tag{5.12}$$

and S_J is the number of bases for a depth J single-tree given by Equation 3.110 in Chapter 3. The D_{J-1}^2 term in Equation 5.11 counts bases from the double-tree grown over the two time halves, and the $S_J - 1$ term includes bases from the single-tree grown over the unsegmented signal. The double-tree algorithm should generate a better optimal basis than the single-tree, because it allows the optimal single-tree to be grown for different segments of a non-stationary signal. The best basis found by the double-tree algorithm must be at least as good as the best single-tree basis, since the set of possible single-tree bases is a subset of the possible double-tree bases.

The double-tree algorithm is asymmetric with respect to time and frequency segmentations. There are many trees over frequency, but only one tree over time, which is grown from the original signal. An obvious extension to the double-tree algorithm is to grow a symmetric tree by allowing binary splits in both time and frequency at each node. This is time-frequency segmentation. Each node in the time-frequency tree has four children, two from a binary time segmentation, and two from a binary frequency segmentation. The original signal is at the root node. At each node, the time-frequency tree is pruned by choosing the winner between time or frequency segmentation, based on the minimum cost. The pruning algorithm starts at the bottom of the tree, and works back to the root to find the optimal subtree. F_J, the number of possible time-frequency bases for a depth J decomposition, is defined by the recursive equation

$$F_J = 2F_{J-1}^2, \tag{5.13}$$

where

$$F_0 = 1. \tag{5.14}$$

F_{J-1}^2 is the number of bases from the time-frequency tree grown over either the time or frequency segment; multiplying by two accounts for the choice between time or frequency. Because the double-tree bases are a subset of the time-frequency bases, the optimal time-frequency segmentation basis must be at least as good as the optimal double-tree basis. The graphical representations of depth two trees in Figure 5.3 compare the single-tree, the double-tree, and the time-frequency tree.

5.4.2 Block time-frequency segmentation

The binary frequency partitioning operation filters a time sequence into lowpass and highpass frequency bands. Because of subsampling by two, each frequency segment has half as many coefficients as the original signal. Binary frequency partitioning can be described as a linear transformation using matrix notation. In a block transform, the filter length is equal to the number of channels. Haar filters used for binary frequency partitioning are an example of a block transform, where both the filter lengths and number of channels is equal to two. For block transforms, there are no edge effects when filtering finite length

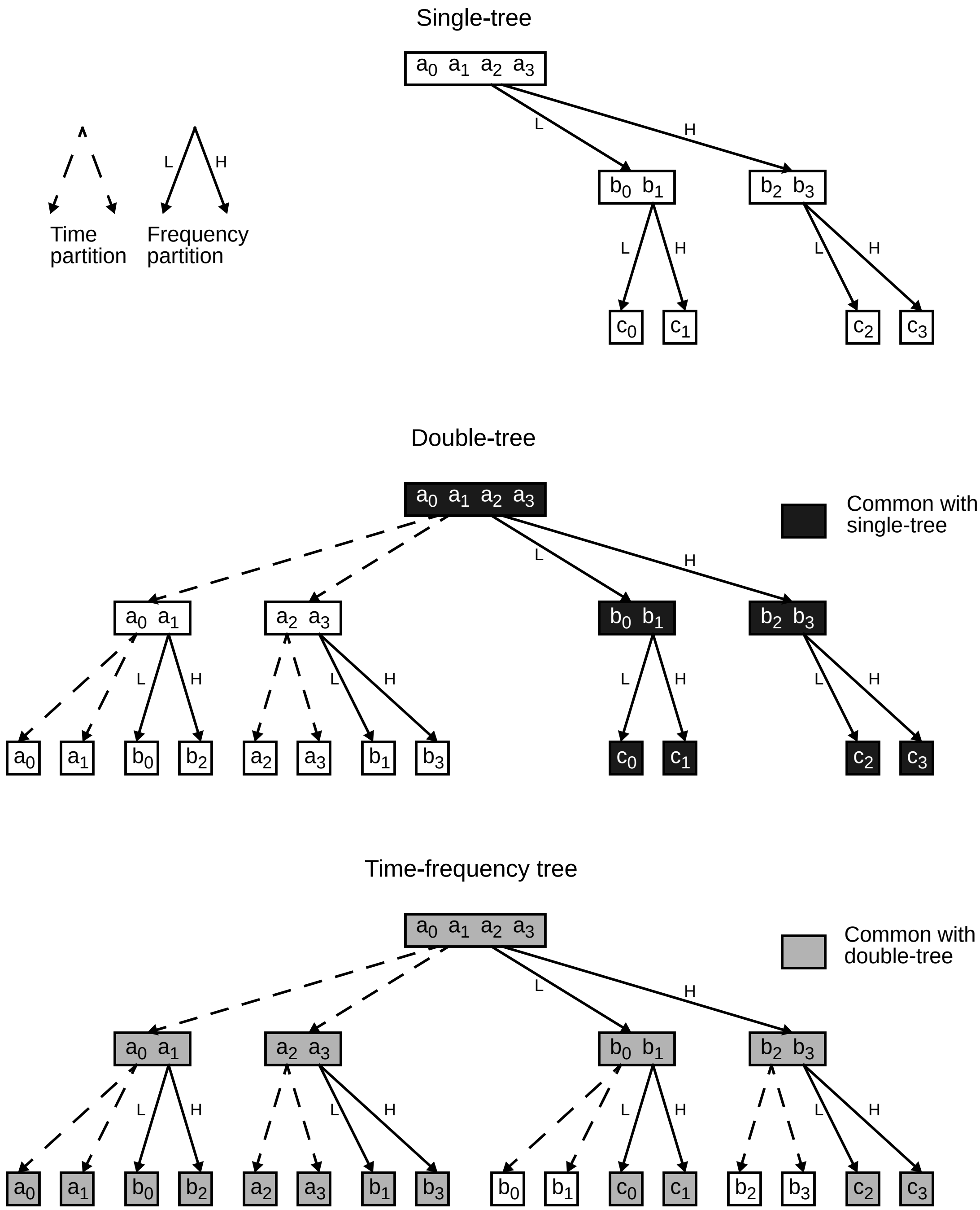


Figure 5.3: Depth two trees

sequences. Because there are no edge effects, time partition followed by frequency partitions is equivalent to frequency partition followed by time partitions.

The block algorithm [27] is an efficient implementation of time-frequency segmentation in terms of both computation and storage. Block time-frequency segmentation searches a subset of the general time-frequency bases, excluding bases that are redundant because of the equivalence of time-frequency and frequency-time partitions for a block transform. The number of possible block time-frequency bases for a depth J decomposition can be obtained from the number of bases for the general time algorithm by subtracting the redundant bases from Equation 5.13, so that the recursive equation becomes

$$F_J = 2F_{J-1}^2 - F_{J-2}^4, \tag{5.15}$$

where

$$F_0 = 1 \tag{5.16}$$

and

$$F_1 = 2. \tag{5.17}$$

Table 5.1 compares the number of possible bases for the single-tree, the double-tree, the time-frequency tree, and the block time-frequency tree. When a block transform is used, the block time-frequency algorithm and general time-frequency algorithm are equivalent. Using a non-block transform (e.g. binary frequency partitioning using filters with lengths greater than two), the block algorithm and general algorithm are close to equivalent if edge effects are insignificant. When edge effects can be ignored, as is usually the case when the filter length is small relative to the length of the sequence being filtered, block time-frequency segmentation can be used to efficiently find a slightly suboptimal time-frequency partition for a non-block transform. Herley et al. report experimental results on the test images Lena and Barb using the Antonini 9-7 filters that show the performance of the block algorithm is very close to that of the general algorithm. All experiments with space-frequency segmentation performed as part of this thesis use the block algorithm.

Table 5.1: Comparison of one-dimensional trees

Depth	Number of possible bases			
	Single	**Double**	**T-F**	**Block T-F**
1	2	2	2	2
2	5	8	8	7
3	26	89	128	82
4	677	8597	32768	11047

The block time-frequency algorithm finds the optimum subtree by building a set of cost trees as it compares the costs between time and frequency blocks. An entry in a cost tree contains a cost and a winning decision, either time or frequency. Cost trees are generated successively. The cost tree at a stage is analogous to the set of pruning decisions at one level of the general time-frequency tree, but excludes all choices that are redundant if time partition followed by frequency partitions is equivalent to frequency partition followed by time partitions. The cost tree at the current stage is used to generate the cost tree at the next stage, which will have one less level than the current cost tree. The final stage cost tree contains a single level, and consists of a single winning decision, the first partition of the optimal partition, and a single cost, the cost of the optimal subtree. The optimal subtree is read out by reversing the trail of winning decisions starting at the final stage cost tree.

A binary segmentation operation divides a segment into two parts. Time segmentation simply divides a sequence in half, while frequency segmentation means lowpass or highpass filtering followed by subsampling by two. The two segments from a partitioning operation can be viewed as subblocks of a block. A frequency block can be viewed as the transformed version of a time block.

The algorithm starts by building the stage zero cost tree. Let J represent the maximum decomposition depth. The stage zero cost tree consists of the rate-distortion cost of each subblock from a wavelet packet decomposition of depth J. At stage zero, the cost tree has $J+1$ levels, and each level has 2^J subblocks. The number of wavelet packet coefficients that correspond to a subblock in the stage zero cost tree is the original sequence length divided by 2^J. The algorithm proceeds by generating cost trees at successive stages. At each level except the last one, the cost of a time block is compared against the cost of the corresponding transformed frequency block, and the minimum cost block is chosen. This rule is the key to the algorithm. The winning cost and winning decision are stored in the next stage cost tree, which has one less level, and each level has half as many subblocks. The cost tree at the last stage has a single level containing a single subblock.

An example partition

Figure 5.4, Figure 5.5 and Table 5.2 illustrate the block algorithm for a depth three decomposition. A subblock is denoted by α_n. The relationship between the subblocks in the stage zero cost tree and the wavelet packet decomposition is shown in Figure 5.4. Figure

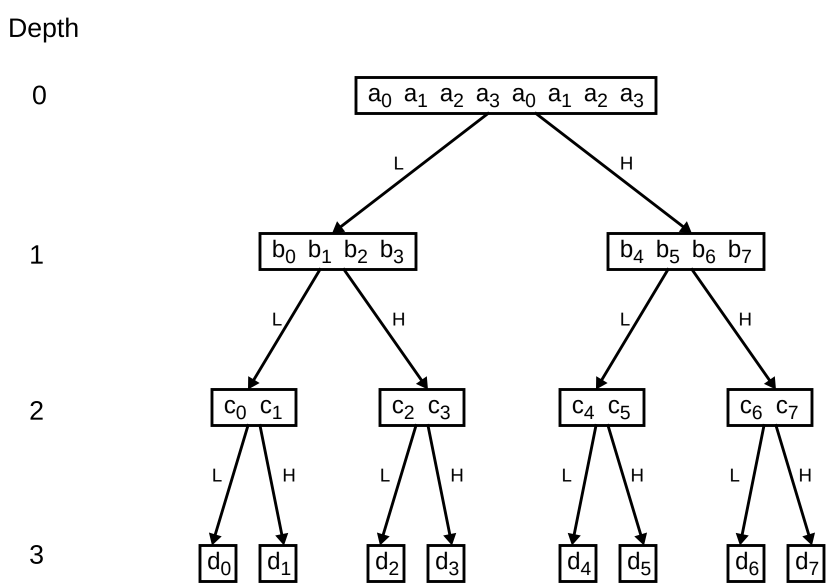


Figure 5.4: Subblocks and wavelet packets

5.5 shows the cost trees generated by the forward algorithm for a depth three decomposition. Table 5.2 shows the grouping of subblocks into time and frequency blocks, comparisons between time and frequency blocks at different stages and levels, and the results

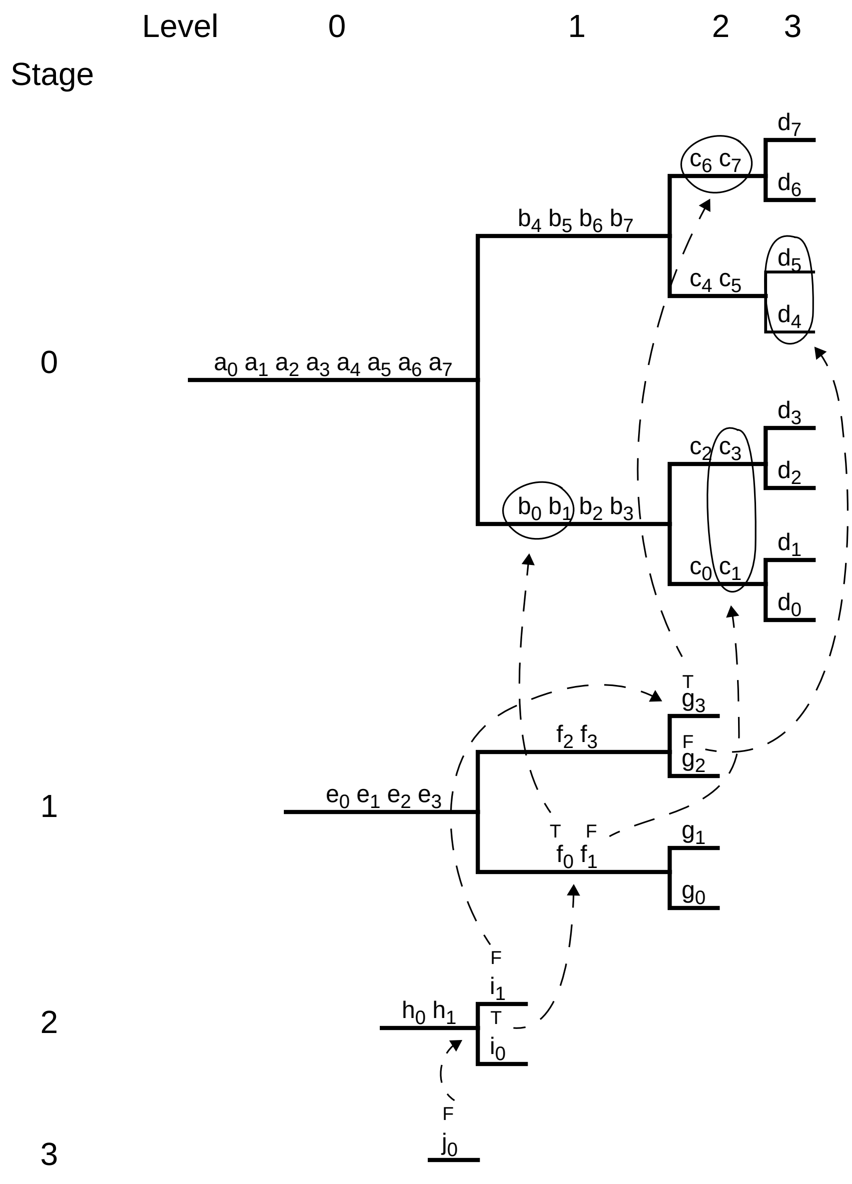


Figure 5.5: Depth three cost trees

Table 5.2: Depth three decomposition

Stage	Level	Block	Subblocks compared		Result
			Time	Frequency	
0	0	0	a_0 a_1	b_0 b_4	e_0
0	0	1	a_2 a_3	b_1 b_5	e_1
0	0	2	a_4 a_5	b_2 b_6	e_2
0	0	3	a_6 a_7	b_3 b_7	e_3
0	1	0	b_0 b_1	c_0 c_2	f_0
0	1	1	b_2 b_3	c_1 c_3	f_1
0	1	2	b_4 b_5	c_4 c_6	f_2
0	1	3	b_6 b_7	c_5 c_7	f_3
0	2	0	c_0 c_1	d_0 d_1	g_0
0	2	1	c_2 c_3	d_2 d_3	g_1
0	2	2	c_4 c_5	d_4 d_5	g_2
0	2	3	c_6 c_7	d_6 d_7	g_3
1	0	0	e_0 e_1	f_0 f_2	h_0
1	0	1	e_2 e_3	f_1 f_3	h_1
1	1	0	f_0 f_1	g_0 g_1	i_0
1	1	1	f_2 f_3	g_2 g_3	i_1
2	0	0	h_0 h_1	i_0 i_1	j_0

stored in the next stage, as the algorithm proceeds. After calculating the subblock costs in the stage zero cost tree from the coefficients of the wavelet packet decomposition, the stage one cost tree can be generated. Level zero of the stage one cost tree is generated by comparing time blocks at level zero against frequency blocks at level one of the stage zero cost tree. The cost of the time block a_0a_1 is compared against the cost of the corresponding frequency block b_0b_4, and the cost of the winner and the winning decision (time or frequency) is recorded in e_0. In a similar manner, the rest of the $a_{2i}a_{2i+1}$ and b_ib_{i+4} subblocks are compared to generate the rest of the e_i subblocks in level zero of the stage one cost tree. The next step is to generate level one of the stage one cost tree. The cost of the time block b_0b_1 is compared against the cost of the corresponding frequency block c_0c_2, and the cost of the winner and the winning decision (time or frequency) is recorded in f_0. The algorithm proceeds through the rest of the $b_{2i}b_{2i+1}$ and c_ic_{i+2} or $c_{i+2}c_{i+4}$ subblocks to generate the rest of the f_i subblocks in level one of the stage one cost tree. The last level in the stage one cost tree, level two, is now generated by comparing the $c_{2i}c_{2i+1}$ and $d_{2i}d_{2i+1}$ subblocks from the stage zero cost tree to generate the g_i. The first entry in level two of the stage one cost tree, g_0, results from the comparison between the cost of the time block c_0c_1 and the cost of the corresponding frequency block d_0d_1. The rest of the stage one, level two cost tree entries are generated in a similar way. There are no more stage zero comparisons to be made, so the stage one cost tree is complete, and the algorithm proceeds to generate the stage two cost tree from the stage one cost tree. Since each entry in the stage one cost tree represents the least cost representation of two subblocks from the stage zero cost tree, subblocks in the stage one cost tree can be grouped into time or frequency blocks and compared in an analogous manner. The algorithm continues until reaching the stage three cost tree, which consists of a single subblock, j_0.

Figure 5.5 also illustrates how the optimum tree is read out of the cost trees. Assume that at the stage three cost tree, the winner at j_0 is frequency. This means that the two winning subblocks in the stage two cost tree are i_0 and i_1, as indicated by a dotted line connecting j_0 to i_0 and i_1. Assume that at i_0 and i_1, the winners are time and frequency respectively. This means that the four winners in the stage one cost tree are f_0 and f_1, and g_2 and g_3. Finally, assume that the winners at f_0, f_1, g_2, and g_3 are time, frequency, frequency, and time respectively. This means that the eight winners in the stage zero cost tree are b_0, b_1, c_1, c_3, d_4, d_5, c_6, are c_7, which defines the optimal decomposition. The final winning subblocks are circled in Figure 5.5.

5.4.3 Extension to two dimensions

Extension of both the general and block time-frequency segmentation algorithms to two dimensions is straightforward. In space-frequency segmentation, binary partition into two one-dimensional segments is replaced by partition into four two-dimensional structures. Space partition divides an image into four quadrants. Two-dimensional frequency partition produces four subbands: LL, LH, HL, and HH. General space-frequency segmentation applies the general time-frequency pruning algorithm to choose between four-way splits in space or frequency in a space-frequency tree. The number of possible bases for a depth J space-frequency decomposition, H_J, is defined by the recursive equation

$$H_J = 2H_{J-1}^4, \tag{5.18}$$

where

$$H_0 = 1. \tag{5.19}$$

Equation 5.18 is similar to Equation 5.13, but squaring is replaced by raising to the fourth power because binary splits in one-dimension are replaced by four-way splits in two-dimensions. Block space-frequency segmentation adapts the block time algorithm to two-dimensional blocks. The principle of comparing the cost of a block to the cost of a transformed block and keeping the winner is unchanged. The number of possible bases for a depth J block space-frequency decomposition, H_J, is defined by the recursive equation

$$H_J = 2H_{J-1}^4 - H_{J-2}^{16}, \tag{5.20}$$

where

$$H_0 = 1 \tag{5.21}$$

and

$$H_1 = 2. \tag{5.22}$$

The subtractive term in Equation 5.20 accounts for sixteen redundant subblocks with H_{J-2} possible expansions. Table 5.3 compares the number of possible bases for the space-frequency tree and the block space-frequency tree. At greater depths, the increase from one to two dimensions results in an enormous increase in the number of possible bases, and makes a fast optimal basis search algorithm a must in the two-dimensional case.

Table 5.3: Comparison of space-frequency trees

	Number of possible bases	
Depth	**S-F**	**Block S-F**
1	2	2
2	32	31
3	2097152	1781506
4	$3.869(10^{25})$	$1.942(10^{25})$

5.5 Space-frequency segmentation and ultrasound image compression

Space-frequency segmentation is a good choice for coding ultrasound images. Rate-distortion optimization is a powerful tool. If the rate and distortion measures are valid, there is no better metric for finding an image representation within the constraints of the chosen quantization and coding scheme. Space-frequency segmentation searches a large library of possible bases for the best representation, and the fast pruning algorithm ensures that the search is performed efficiently. Ultrasound images have some features that distinguish them from the normal images that are the usual targets of compression algorithms. Space-frequency segmentation finds the rate-distortion optimal representation for a particular image without any underlying assumptions about the content of the image. The optimal space-frequency partition can also provide some insight into the nature of ultrasound images, which can be used to design a better codec.

Chapter 6

Results

6.1 Introduction

The goals of the experiments are to identify and understand the important parameters of a space-frequency codec, to select a set of good parameters for the coding algorithm, and to evaluate the use of space-frequency segmentation to code ultrasound images. Because there are no widely accepted standard medical ultrasound test images, images coded using space-frequency segmentation are compared against images coded using SPIHT, the wavelet transform based zerotree codec of Said and Pearlman. The subjective quality of the compressed images in terms of medically relevant features is assessed by ultrasound radiologists. Finally, the optimum partition of a set of ultrasound images is investigated, to find a common fixed partition that works well, and to better understand the nature of ultrasound images.

Note that this research does not fully optimize the codec in that there are still many parameters to choose and vary. Some of these parameters are the entropy coding scheme, the maximum decomposition depth, the filters, and the quantizers. Scalar quantizers are used throughout the experiments, and the quantizers also have numerous parameters that can be varied, including the number of quantizers, the step size of each quantizer, the alphabet size of each quantizer, and the granular range that each quantizer spans. The choice of a good set of parameters is further complicated by the interaction that is possible between parameters, such as the use of different quantizers and entropy coding schemes for different groups of subbands, or the use of different filters at different decomposition depths. The goal of the experiments reported in this chapter is to evaluate the potential of space-frequency segmentation for the compression of medical ultrasound images. To this end, a good set of codec parameters is determined experimentally, then applied to the compression of ultrasound images. Exhaustive investigation of space-frequency codec configurations to find the best codec is a task for future research.

6.2 Codec details

6.2.1 Implementation details

Before any other operations are performed, the original image is preprocessed by subtracting an offset value, in order to roughly center the coefficient distribution around zero. Centering the distribution allows the use of symmetric quantizers. For simplicity and to avoid sending the offset value as side information, the value is derived from the precision. For an image with a precision of n bits, the offset value is 2^{n-1}. An 8-bit image with a possible pixel range of 0 to 255 is converted to have a possible pixel range of -128 to +127.

The space-frequency encoder and decoder are implemented as separate programs, which communicate only through a file containing the compressed image representation. This means that all coded rates reported for the implemented space-frequency segmentation codec are reliable, and are calculated from the size of the representation file. Like a raw image representation, the file does not include the image width and height, but does include the side information specifying the optimum partition and the quantizer for each subband, in addition to the quantized subband coefficients.

The block space-frequency segmentation algorithm is implemented instead of the full algorithm. Herley et. al. [27] report results for both algorithms which show that the block algorithm is almost as good as the full algorithm. Unless explicitly stated, the 9-7 biorthogonal filters of Antonini et. al. are used, and symmetric extension is applied to handle block boundaries. Use of these 9-7 filters is very common in the source coding literature.

6.2.2 Subband classifications

Subbands are classified into three groups according to the distribution of their coefficients: high pass bands, low pass bands, and space-only bands. The codec treats subbands differently according to their classification. Low pass subbands are those which have been generated without any high pass filter applications. Low pass subbands look like subsampled versions of the image prior to filtering, but with increased dynamic range. The coefficient distribution has similar shape to that of the unfiltered image, but repeated low pass filtering followed by subsampling concentrates energy into fewer and fewer coefficients, increasing the magnitude of individual coefficients and increasing the range of coefficient values. High pass subbands include all subbands that have been filtered with the high pass analysis filter at any point in their decomposition hierarchy. High pass subbands have less energy, and have a large number of zero valued coefficients. Space-only subbands are partitioned only in space; they are not filtered or subsampled at all. Coefficients in space-only subbands retain the characteristics of a rectangular portion of the original unpartitioned image. To illustrate the different subband classifications, Figure 6.1 shows the coefficient distribution of Lena and the ultrasound only image Ub (which was used in Chapter 4), and the distributions of the LL and HH subbands after depth one frequency decomposition for both images. In the figure, the coefficient counts of the original image are scaled by a factor of $^1\!/_4$ to allow comparisons across scales, and the coefficient counts of the HH band are scaled by a factor of $^1\!/_{10}$ to display all three distributions clearly. Note that the coefficient distribution of Ub shows a large spike at the minimum pixel value, and is skewed towards black compared against the distribution of Lena. As long as the skew does not cause excessive overload distortion, it is not a problem, since the quantized coefficients are entropy coded.

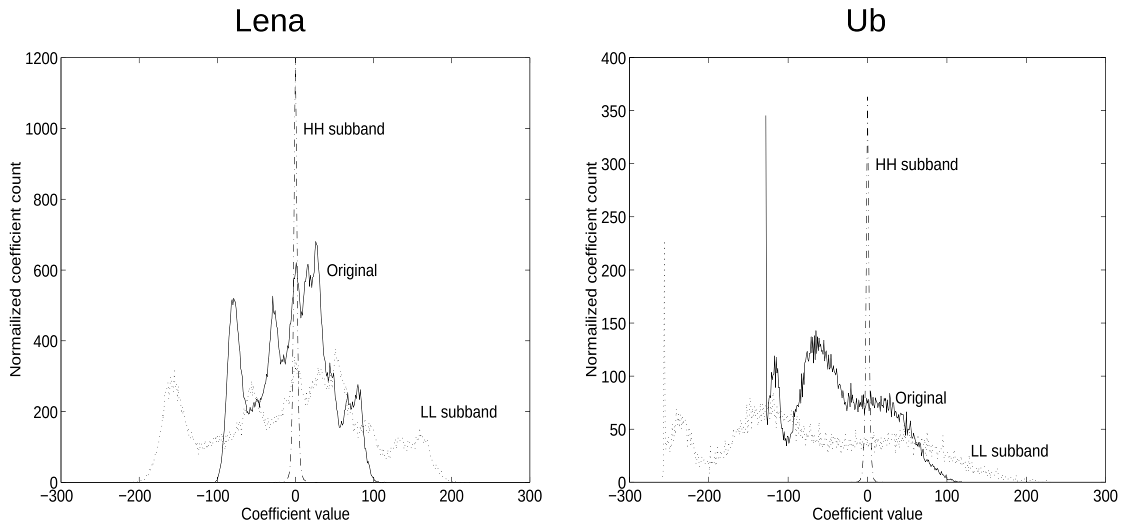


Figure 6.1: Coefficient distributions for Lena and Ub

A subband is quantized with a scalar uniform quantizer, because scalar quantization followed by entropy coding is simple and generally works well. Each quantizer is defined by its alphabet size, step size, and range. The choice of quantizer is an important parameter in determining the performance of the codec. Depending on the classification of the subband, the quantizer for a particular subband is chosen from a different collection of quantizers. There is no additional computational cost in allowing different groups of possible quantizers for the three subband groups.

Each subband group is allowed a different method of coding, and a different estimate of rate in the rate-distortion cost calculation. Choice of coding method allows the best choice to suit the characteristics of subbands in a particular group. For example, a coding method that efficiently codes zero value coefficients is well suited to high pass subbands, but less suitable for low pass or space-only subbands. The computational cost is high when the rate used in the best partition search is calculated by actually coding the quantized coefficients. Separating the rate calculation and coding scheme allows an estimate of rate that is computed quickly to be used instead of the actual coded rate, as long as the estimate is consistent with the coded rate. One example is to use zeroth-order entropy to estimate the rate, then code the subband coefficients with an arithmetic coder. Using an estimate instead of the actual coded rate produces a sub-optimal partition, but the increased speed may outweigh the loss in fidelity. Again there is virtually no additional computational cost in allowing different rate calculation and coding schemes for different subband groups.

6.2.3 Verification of implemented codec integrity

Results from the implemented codec are compared with Herley's published results using a single quantizer step size for all subimages, to verify the sanity of the implementation of the block space-frequency algorithm. The test images Lena and Barb are compressed at 1.0 bpp, 0.5 bpp, and 0.25 bpp, and the compressed image quality is reported in terms of PSNR. In each case, the single step size is chosen to meet the target rate. Table 6.1 compares the implemented experimental performance against the published performance. The data in the table shows good agreement between the reported results and the results from the implemented codec. Note that all rates in Table 6.1 are zeroth-order entropy estimates (as in Herley's paper), not actual coded rates.

Table 6.1: Herley's published results and results from the implemented codec

	Herley published		**Implemented codec**	
Image	**R (bpp)**	**PSNR (dB)**	**R (bpp)**	**PSNR (dB)**
Lena	1.0	39.77	0.9966	39.57
Lena	0.5	36.64	0.4999	36.62
Lena	0.25	33.51	0.2498	33.52
Barb	1.0	36.73	0.9967	36.75
Barb	0.5	32.07	0.4998	32.06
Barb	0.25	28.31	0.2495	28.42

6.2.4 Partitioned image representation

An image of subbands is a useful tool for studying the optimum space-frequency partition. Figure 6.2 shows partition of the test image Lena into space and frequency subbands.

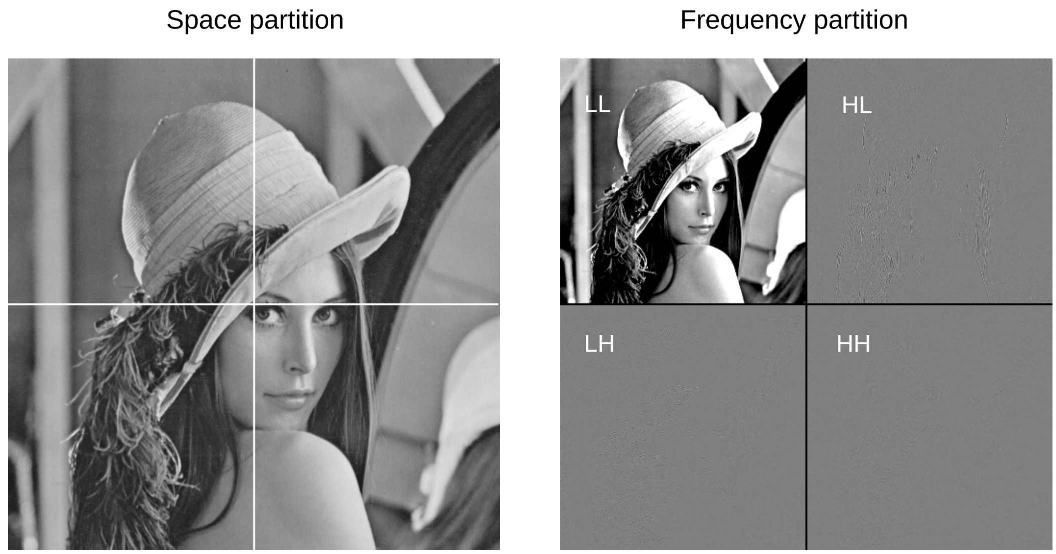


Figure 6.2: Subband partitions

Since both space and frequency partitions occur in a space-frequency segmented subband image, space partitions are denoted by a pair of intersecting white lines that separate a band into four quadrants, and frequency partitions are denoted by a pair of intersecting black lines that separate a band into four frequency subbands.

As an example, a space-frequency decomposition along with the corresponding subband image is illustrated in Figure 6.3, again using Lena. The maximum depth of the example decomposition is two. The first partition is frequency, then the LL band is partitioned in frequency again, and the LH and HL bands are partitioned in space, but the HH band is not partitioned any further.

6.3 Coding space-frequency segmented images

There is a trade-off between efficient coding and efficient quantization of space-frequency segmented images. Knowledge of the source probability density function is required by

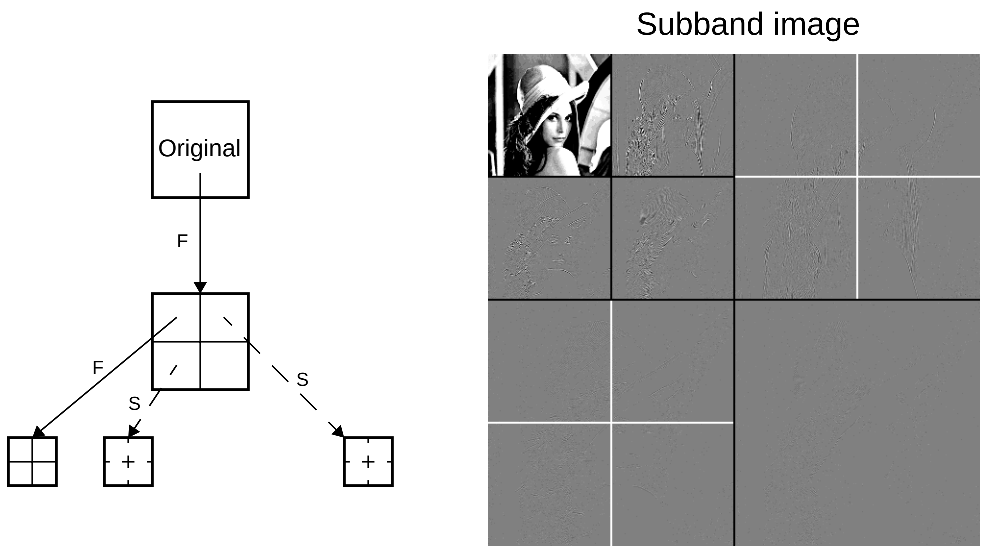


Figure 6.3: Example space-frequency decomposition

the entropy coder. Statistics can be gathered at the encoder by making a pass through the data, but the information must then be sent as side-information to the decoder. When coding many subimages, the extra side-information required to describe the statistics for each sub-image can be a significant amount of overhead. If the encoder and decoder are adaptive, no side-information describing source statistics needs to be sent. An adaptive coder learns on the fly and can deal with changing source statistics, but how well the coder can adapt depends on the alphabet size and sequence length. A small alphabet combined with a large sequence means that the coder acquires a good estimate of source statistics relatively quickly, and produces an efficient code.

Doing a space partition does not change the underlying coefficients, but allows different quantizers to be used on different areas of the same region. The freedom to use more quantizers is generally advantageous, since the best quantizers are chosen to suit different statistics in different regions. In a space-frequency segmented image, it makes sense to code subbands separately, however as decomposition depth increases, subbands become smaller and are coded less efficiently. An additional complication is that a coding scheme that exploits intra-subband dependance may prefer coding larger subbands. The trade-off is between using many quantizers and coding smaller subbands less efficiently, or using fewer quantizers for the same region but coding the larger resulting subbands more efficiently. A simple solution is to assume that the coder always works well, and quantize and code the minimum size subimages dictated by the decomposition depth. For any combination of space and frequency partitions, the minimum size subimage equivalent is obtained by additional space partitions to the maximum depth without any more frequency partitions. Herley et. al. [27] describe this approach, and by using entropy to estimate coded rate but not actually coding any images, they completely avoid the problem of coding small subbands. Letting the rate-distortion optimization process decide the resolution at which to code and quantize subbands produces the best trade-off, at the expense of increased computation.

Another fundamental question is which entropy coding scheme to choose, in order to balance coding efficiency against simplicity and fast computation. Arithmetic coding (discussed in Section 3.2.2) is one good choice. When the alphabet is relatively small and the sequence to be coded is relatively large, an arithmetic coder operating on individual coefficients can achieve a code rate close to the zeroth-order entropy. A coder that exploits correlation between coefficients can do better than the zeroth-order entropy. Stack-run coding (discussed in Section 3.2.3) is a simple technique that combines run-length coding with adaptive arithmetic coding on a small alphabet. For certain sources that produce many zeros, stack-run produces a code rate that is lower than the zeroth-order entropy. Other schemes are possible as well, but generally with increased complexity.

6.3.1 Coding experiments using natural images

Test images are required in the study of image compression. In the source coding literature, a small set of test images is generally used, permitting comparison of results using different source coding techniques. The test images are chosen so that their content is representative of commonly encountered scenes. Two of these images, Lena and Barb, have already appeared in this thesis. The label “natural images” is used throughout this thesis to refer to such images, and also to distinguish them from ultrasound images. While they do not represent all images, the “natural images” used in this thesis are chosen to represent a broad class of images (e.g. a person against a background), and to permit the results of this thesis to be assessed in the context of the body of source coding literature.

The investigation of entropy coding schemes and space-frequency segmentation is carried out using natural images. Natural images are chosen so that the results can be discussed in the context of Herley et. al.’s original work, since they do not investigate the use of actual entropy coders. Use of natural images also allows the space-frequency technique to be evaluated against other image compression techniques in a fair comparison. By using natural test images, the focus of the coding experiments is the interaction between the coding scheme and the space-frequency segmentation algorithm, which tells how to effectively code space-frequency segmented images before proceeding to the task of compressing ultrasound images.

Test images

The test images chosen to investigate coding performance are 512×512 versions of Lena and Barb[1]. Both images are commonly used to evaluate lossy compression schemes. Lena is relatively easy to code, but Barb has some features that make it more difficult to compress, particularly the fine striped pattern on the trousers and shawl. The target bit rate for investigating coding schemes is chosen to be 0.5 bpp. Based on visual assessment of Lena and Barb compressed using the Said and Pearlman SPIHT codec, 0.5 bpp is a reasonable target rate. At 1.0 bpp the compressed image is virtually indistinguishable from the original, especially in the case of Lena. Artifacts are clearly visible at 0.25 bpp, and the image quality is generally poor. At 0.5 bpp, compression artifacts are present but are not severe, making it an interesting target bit rate.

1. The RPI version of Lena (ftp://ipl.rpi.edu/pub/image/still/usc) and the Gersho version of Barb are used for the experiments in this chapter.

Maximum decomposition depth

Five is chosen as the maximum space-frequency segmentation decomposition depth. Up to a certain point increasing the decomposition depth improves the compression performance, since there are more bases to choose from. Obviously, the depth is limited by the ability to recursively sub-divide the image into four equal size subimages. There are several other factors which practically limit the depth, however: the expansion of the number of possible bases, the validity of the block algorithm assumptions, the efficiency of coding small subbands, and the increase in side information. Since the number of possible bases increases exponentially with depth, at some point the computational cost outweighs the improvement in compression with increasing depth. Another factor that limits the maximum depth is the assumption that edge effects from filtering can be ignored, which allows the block space-frequency algorithm to find a partition which is nearly optimal. Edge effects can be ignored when the filter length is short with respect to the length of the sequence being filtered: starting with a 512×512 image and using 9-7 filters, at a depth of five a filter of maximum length nine is applied to a sequence of length thirty-two. As depth increases, sub-images become smaller, and small subimages cause problems for adaptive entropy coding because the coder has a smaller sequence over which to obtain a good estimate of the statistics. Another problem with increasing depth is the increase in side information. The optimum partition description is negligible, but the quantizer description for each subband is not. If all subbands are minimum size, there are 1024 subimages from a depth five decomposition. If there are sixteen possible quantizers, a 4 bit quantizer description for each subimage takes up over 3% of the bit budget of a 512×512 image compressed at 0.5 bpp. Trading off all these different factors, a maximum depth of five is a reasonable choice.

Quantizer sets

The initial task is to find a reasonable set of uniform scalar quantizers to use with the test images at the target rate. The goal is to use the same set of quantizers for both images. In an actual codec, different quantizer sets for different images is impractical, though different quantizer sets for different classes of images is certainly feasible (e.g. natural and ultrasound images). Since Lena and Barb represent an easy and a harder case for normal images, using one quantizer set for both images suggests that a single quantizer set can work for a range of images.

For each subband, there is a choice among sixteen different quantizers. More quantizers allows a better quantizer choice to be made, but computational cost increases linearly with the number of quantizers, and the side information rate also increases when more quantizers are used. The choice of sixteen is a reasonable trade-off, and Herley et. al. [27] also report results using sixteen different quantizers.

The alphabet size for space only bands is determined by the step size, and is chosen to cover the possible pixel range after preprocessing, -128 to +127. The alphabet size for the low pass and high pass subbands is sixty-four, and is chosen small enough to allow reasonably efficient coding of small bands (at depth five, the smallest subimage will contain 256 coefficients), but large enough to ensure that overload distortion is not a problem. The alphabet size can be tailored to suit the size and distribution of subimages at different levels, but this adds too much complexity and is not attempted.

The primary variable defining a scalar quantizer is the step size. The method used to determine a good step size range for the quantizer sets is to start with a large spread in step size, and to refine the range by observing histograms of quantizer usage. A good step size range is found for quantizer sets for Lena and Barb, using entropy, arithmetic coding and stack-run coding, and restricting subbands to the minimum size or allowing the optimization to determine the best subband size. Experiments are also conducted that vary the step size linearly, or exponentially at a constant rate over the step size range, using entropy, arithmetic coding and stack-run coding. Virtually no difference is found by varying the step size linearly or exponentially, so a linear step size variation is chosen since it is intuitively simplest. For some subband groups, including a single coarse quantizer (with step size 128) outside of the linear variation range is determined experimentally to be effective, so only fifteen out of the sixteen total quantizers for these groups have a linear step size variation. The information from these experiments is combined to set the step size ranges for the common quantizer set, which is also tested with entropy and the different coding schemes. The results with the common quantizer set are almost as good as the best quantizer set for a specific image and coding scheme, so the common quantizer set is used for subsequent coding experiments. The step size, alphabet size, and minimum output value for each quantizer in the common quantizer set are given in Appendix A.

Subimage size

Calculation of rate-distortion costs is the critical path in the space-frequency segmentation algorithm. The optimal space-frequency partition can be viewed as a tree structured combination of space and frequency partitions that is constrained by a maximum decomposition depth. Each node in the decomposition tree corresponds to a subimage. When the optimization determines the optimal subimage size, there is a three way decision at each node that is not at the maximum depth: partition in space, partition in frequency, or no further partition. This three way decision requires three corresponding rate-distortion costs. Note that space partition does not change the underlying coefficients. The choice of no further partition at a non-maximum depth node can be eliminated without reducing the basis choices, because multiple space partitions without any further frequency partitions can be applied to any combination of space-frequency partitions to produce minimum size subimages without altering the underlying coefficients. To illustrate the concept of minimum size subimages, Figure 6.5 shows the subband image for the depth three wavelet transform of Lena, and the minimum size subimage equivalent. When minimum size subimages are used, only the rate-distortion costs of subimages at maximum depth need to be computed. The significant reduction in the number of costs that need to be calculated produces a corresponding significant reduction in the time it takes to find the best partition.

From a quantization point of view, it is obviously equivalent to quantize a region with a single quantizer, or to quantize multiple space partitions of the same region with that same quantizer. In fact, minimum size subimages mean that the best quantizer is always chosen for a region at the minimum resolution allowed by the depth constraint.

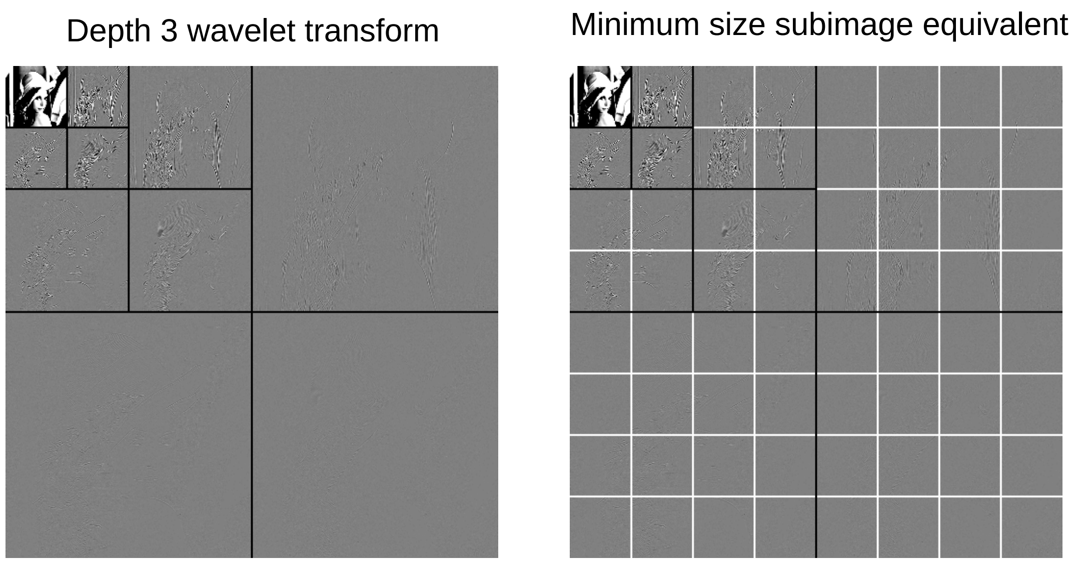


Figure 6.4: Wavelet transform and minimum size subimage equivalent

The only penalty in terms of quantization for the use of minimum size subimages is that the maximum amount of side information to specify the quantizers is always required.

The real problem with minimum size subimages is the difficulty that a practical coder has in efficiently coding small subimages. An adaptive entropy coder incurs a higher relative penalty for learning the statistics of a shorter sequence, and a run-length coder is limited to a shorter sequence over which it can code runs of identical values. Since zeroth-order entropy as a measure of rate does not account for any of the problems faced by a practical coder, experiments will show that it is an optimistic measure of rate, which does not illuminate any of the coding problems associated with using minimum size subimages.

The best method to choose subimage size to trade-off coding and quantization efficiency is investigated. Using minimum size subimages or allowing the optimization to determine the size of each subimage, Lena and Barb are compressed using space-frequency segmentation at a target rate of 0.5 bpp. The actual rate using arithmetic coding for all three subimage groups is compared against the entropy estimate of rate, since the results should be the same if subimage size is not an issue for coding. Table 6.2 reports the experimental results. The experiments are performed on a Pentium II Xeon system running Windows NT 4.0.

Table 6.2: Space-frequency segmentation and subimage size

Image	Coding scheme	Subimage size	Enc time (min:sec)	R (bpp)	PSNR (dB)
Lena	Entropy	Minimum	0:09	0.4983	36.87
Lena	Entropy	Optimization determined	0:22	0.4983	36.92
Lena	Arithmetic	Minimum	0:42	0.4995	35.77
Lena	Arithmetic	Optimization determined	1:24	0.4972	36.29
Barb	Entropy	Minimum	0:09	0.5018	32.47
Barb	Entropy	Optimization determined	0:22	0.4997	32.48
Barb	Arithmetic	Minimum	0:42	0.5011	30.99
Barb	Arithmetic	Optimization determined	1:24	0.4975	31.67

The encoding time when the optimization determines the subimage size is approximately twice the time required when all subimages are minimum size, when entropy is used as an estimate of rate and when subimages are actually coded using arithmetic coding. This confirms that use of minimum size subimages is very effective in speeding up the space-frequency codec. As expected, subimage size is not a performance issue when using entropy to estimate rate; using minimum size subimages and using optimum size subimages produce virtually identical PSNR. Note that this indicates that in the case of zeroth-order entropy, the difference in side-information between minimum size and optimal size subimages trades off against the freedom to choose the optimal subimage size to produce very similar results. Clearly different results between minimum size subimages and optimum size subimages are produced with arithmetic coding. When the optimization determines the subimage size, the PSNR is 0.68 dB better for Barb and 0.52 dB better for Lena. In the optimum size subimage case, there are many large subimages which show variation in content across the subimage, but are not further partitioned in space. Herley et. al. restrict the subimages to minimum size, but because they use zeroth-order entropy to estimate the rate instead of actually coding the subimages, their results are misleading.

Entropy as a measure of rate is optimistic, because it ignores the problem of coding small subimages. The experiments show that for a real coder, subimage size is a problem for efficient coding, and allowing the optimization to choose the subimage size is the best way to trade off the conflicting demands between quantization and coding. Even though the penalty in terms of computation time is significant, the performance cost is too high to justify the use of minimum size subimages.

Entropy coding techniques

Space-frequency segmentation together with stack-run coding is used to compress Lena and Barb at 0.5 bpp. Because stack-run coding requires energy compaction and a high concentration of zero-valued coefficients, it is only applied to the high pass bands. Arithmetic coding is used to code the low pass and space-only bands. The size of each subband is chosen by the rate-distortion optimization algorithm. Table 6.3 compares stack-run coding against SPIHT, and against arithmetic coding all subbands (repeating the arithmetic coding results from Table 6.2). In terms of PSNR, space-frequency segmentation is better

Table 6.3: Lena and Barb compressed at 0.5 bpp

Image	Codec	R (bpp)	PSNR (dB)
Lena	SF - stack-run	0.4962	36.42
Lena	SF - arithmetic	0.4972	36.29
Lena	SPIHT	0.500	37.21
Barb	SF - stack-run	0.4961	31.92
Barb	SF - arithmetic	0.4975	31.67
Barb	SPIHT	0.500	31.40

than SPIHT at compressing Barb, but is worse than SPIHT at compressing Lena. The subjective quality of the compressed images appears consistent with the PSNR. Details in Barb are preserved slightly better by the space-frequency compressed version, while the low frequency areas of Lena look better in the SPIHT version. The subband images for Lena are shown in Figure 6.5, and the subband images for Barb are shown in Figure 6.6. Since the rate-distortion optimal space-frequency partition of Lena is quite wavelet transform like, a coding scheme like SPIHT which uses the wavelet transform and exploits both inter-band and intra-band correlation is expected to work well. The optimal partition for Barb is less similar to the wavelet transform, and shows more frequency decompositions in vertical subbands, due to the stripes on the trousers and shawl.

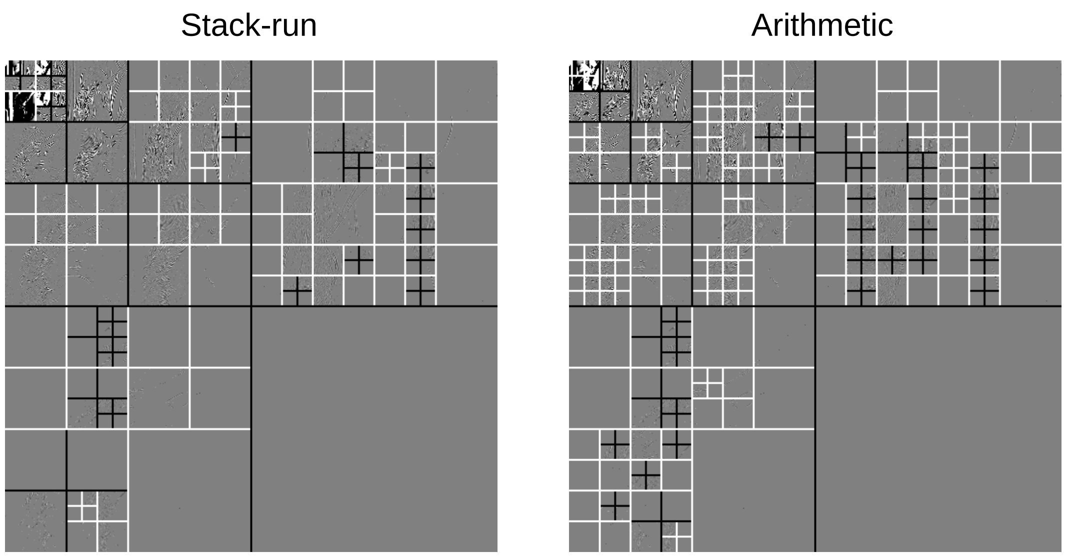


Figure 6.5: Lena subband images

On both Lena and Barb, stack-run coding the high pass bands performs better than arithmetic coding the high pass bands. Since stack-run coding exploits inter-band correlation, it is expected to work better, but the difference in terms of PSNR is not that large (0.13 dB for Lena, 0.25 dB for Barb). Again, the optimum partition explains why stack-run coding is not dramatically superior. Compare the arithmetic case against the stack-run case in the subband images of Lena and Barb. The larger the subimage, the more efficient stack-run coding is over arithmetic coding. The larger subimages required for coding efficiency in the stack-run case limit the freedom to use many quantizers. Small space partitioned subbands in the arithmetic coding case indicate the use of many different quantizers, but the better quantization choices are balanced against the use of a less efficient coding scheme. The power of the space-frequency segmentation algorithm is evident, since it applies the strengths of either of the two different coding schemes to produce

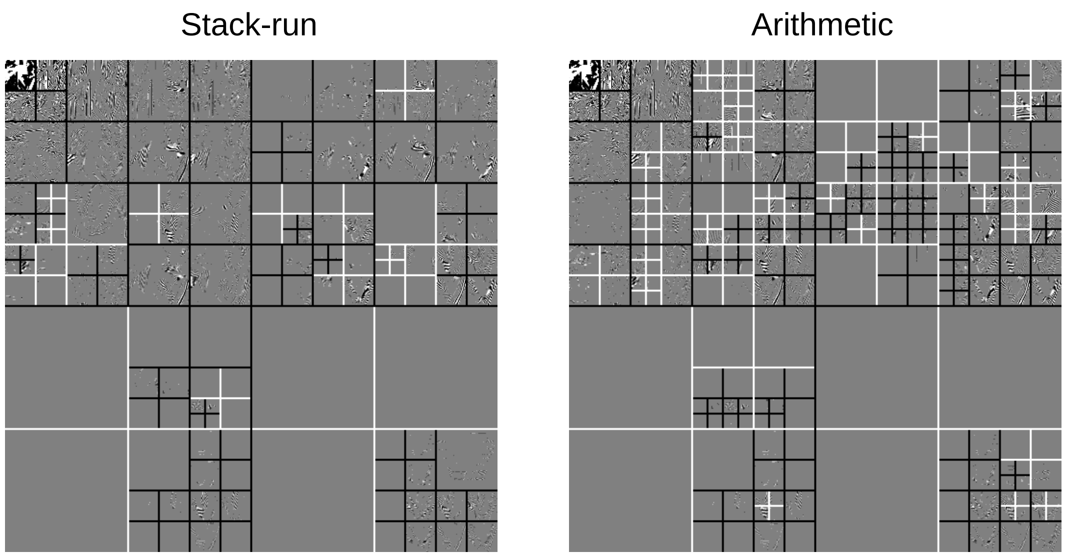


Figure 6.6: Barb subband images

quite similar PSNR. The two different coding schemes do not differ much in computation time; the extra step of forming stacks and runs followed by arithmetic coding a small alphabet balances out against arithmetic coding a large alphabet. Since there is not much difference in computational cost, stack-run coding is preferred, since in theory it can do better than the zeroth-order entropy, even though in practice the improvement over arithmetic coding is not as large as hoped for.

6.4 Compressing ultrasound images

6.4.1 Ultrasound test images

For manageability, two test images are chosen for the ultrasound compression experiments. Both test images are 640×480 pixels in size. An image consists of an ultrasound scanned area against a dark background, with some light colored text in the background area describing the exam conditions. A bar of grayshades provides a visual indication of the grayscale range, and appears in the left side of both images. The ultrasound area of both images is wedge shaped, implying that curved linear array transducers have been used to capture both images. Abdominal exams are a common application for diagnostic ultrasound, and both test images are of the liver. Both images are quite speckly, which is typical for liver images, but it is not clear whether or not the speckle is diagnostically significant. (For an example of diagnostically significant speckle in liver images, see the paper by Krasner et. al. [31] reviewed in Section 4.1.) The text portions of both images contain a small section of inverse video (e.g. dark text on a light background).

The two test images are chosen to represent an easy and a harder case for compression, while still being typical images. The test images are shown in Figure 6.7. The harder case, labelled U1[2], is an image of a normal liver captured at 3.5 MHz. U1 shows high contrast and significant fine detail, particularly in the upper central area. There is also low resolution detail present in the low contrast dark areas across the center of the image. An unusual feature in U1 is the icon in the upper left indicating the organ being scanned. The easy case to compress case, labelled U7[3], shows a lesion in the liver. U7 was captured using a 5 MHz transducer. At the time of capture the contrast was adjusted to highlight the lesion, so the overall image is quite dark and low contrast. The ultrasound area of U1 occupies a larger proportion of the total image than the ultrasound area in U7. A radiologist commented that U1 is quite a typical image in terms of quality and content, but a bit higher contrast than usual, and containing significant textural detail. The radiologist viewed U7 as a poorer quality image without a lot of detail, but since the loss of detail results from the intentional highlighting of the lesion, it is not atypical.

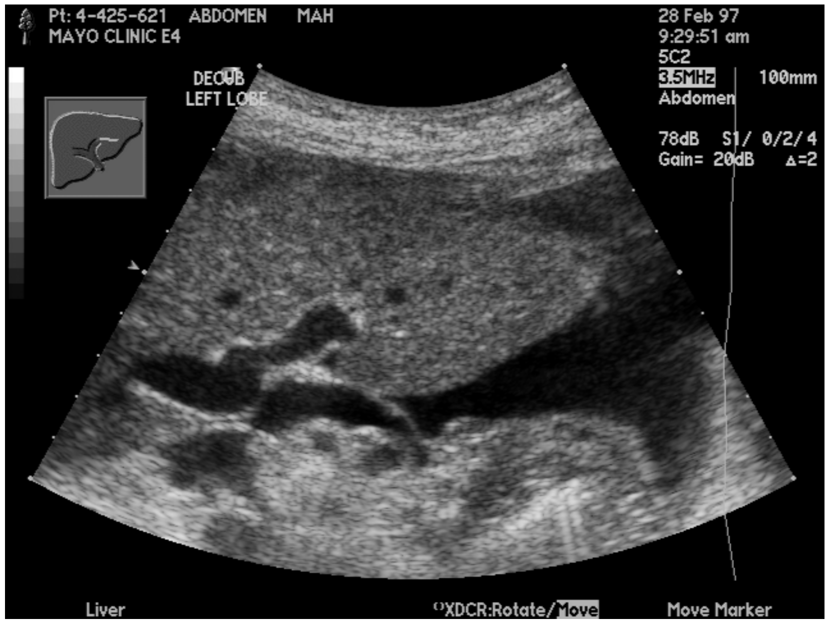
6.4.2 Compression experiments: image quality at 0.5 bpp

The choice of and justification for the codec parameters for the compressed image quality experiments are similar to those from the coding experiments on normal images. The target bit rate is 0.5 bpp, and is chosen based on subjective image quality such that artifacts

---

2. U1 is obtained from the ALI Technologies website, www.alitech.com. ALI is a manufacturer of medical PACS
3. U7 is obtained from the Radiology Department of Vancouver General Hospital.

U1



U7

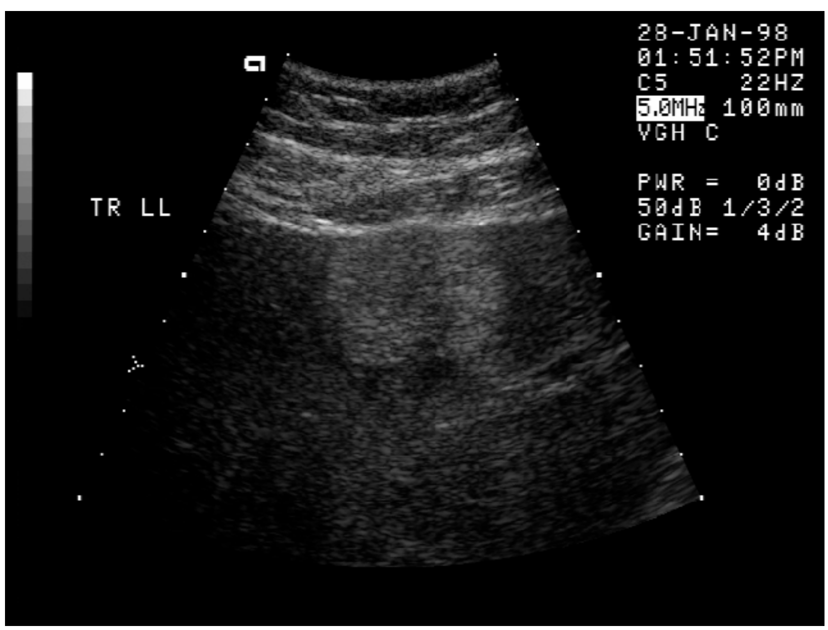


Figure 6.7: Ultrasound test images

are visible but image quality is still quite high. A radiologist who examined U1 and U7 compressed at 0.5 bpp using space-frequency segmentation confirmed the choice of target bit-rate, commenting that artifacts in U1 are present but image quality is still quite good, while artifacts in U7 are hard to detect, and the slight edge sharpening around the lesion in the compressed version of U7 actually makes the compressed version preferable to the original. The maximum decomposition depth is five, since 2^5 is the largest power of two factor of the image height of 480 pixels, and since the use of the Antonini 9-7 filters means that a filter of maximum length nine is applied to a sequence of minimum length thirty at depth five. For each subband there is a choice between sixteen different quantizers, and high pass, low pass, and space-only subbands choose from a different collection of quantizers. A common quantizer set is used for both U1 and U7. The alphabet size of the quantizers for the space-only bands is chosen to cover the possible pixel range of the preprocessed image, -128 to +127, and the alphabet size for quantizers for low pass and high pass bands is sixty-four. The step size, alphabet size, and minimum output value for each quantizer in the common quantizer set are given in Appendix B. High pass subbands are stack-run coded, while low pass and space-only subbands are arithmetic coded.

Using squared error as an objective measure of compressed image quality, space-frequency segmentation performs better than SPIHT for both test images at 0.5 bpp. Table 6.4 reports the PSNR from space-frequency segmentation and from SPIHT. The improvement of space-frequency segmentation over SPIHT is 0.73 dB in the case of U1, and 1.50 dB in the case of U7.

Table 6.4: PSNR for U1 and U7, 0.5 bpp

Image	**Codec**	**R (bpp)**	**PSNR (dB)**
U1	Space-frequency	0.5008	32.9910
U1	SPIHT	0.500	32.2619
U7	Space-frequency	0.4991	38.6024
U7	SPIHT	0.500	37.1035

One of the problems in using SPIHT to compress ultrasound images is that degradation of subjective quality is quite noticeable even at high PSNR values. Because it allocates more bits to large magnitude coefficients in low frequency bands, SPIHT seems to have trouble with the speckle and other fine details in ultrasound images. The subjective quality of the space-frequency compressed versions are superior to the SPIHT compressed versions for both test images, and the overall subjective quality of the space-frequency compressed images is quite high when compared against the original versions. Since the coding rate is chosen so that the recovered estimate is approximately diagnostically equivalent to the original, it is difficult to see the difference between recovered estimate and original images reproduced at low resolution on paper, so pictures of the recovered estimates are not included here.

Comparing the space-frequency compressed version of U1 to the original, there is some blurring of fine high contrast textural detail in the compressed image, and there is some loss of low contrast detail in the dark areas across the center of the image. An interesting artifact that is present in the space-frequency compressed U1 but not the SPIHT compressed U1 is the faintly visible boundary between space partitioned regions. The boundary is most noticeable for space partitions that occur early in the decomposition hierarchy. This boundary artifact is analogous to the blocking effect in transform coded images, and is only produced by space partitions. Comparing the space-frequency compressed version of U1 to the SPIHT compressed version of U1 shows that less detail is preserved by SPIHT, and more artifacts are introduced by SPIHT. Figure 6.8 zooms in on a region of fine detail in U1, comparing the original, space-frequency compressed version, and SPIHT compressed version. Note the vertical space partition boundary visible in the space-frequency compressed version, about $^1/_3$ of the way across the zoomed area, and the artifacts and blurring in the SPIHT compressed version. Figure 6.9 zooms in on a region of low contrast detail in U1, comparing the original, space-frequency compressed version, and SPIHT compressed version. Note the blurring of low contrast detail in the SPIHT compressed version.

The improvement in subjective quality of space-frequency segmentation over SPIHT is less noticeable for U7 than U1, since the characteristics of U7 make it easier to

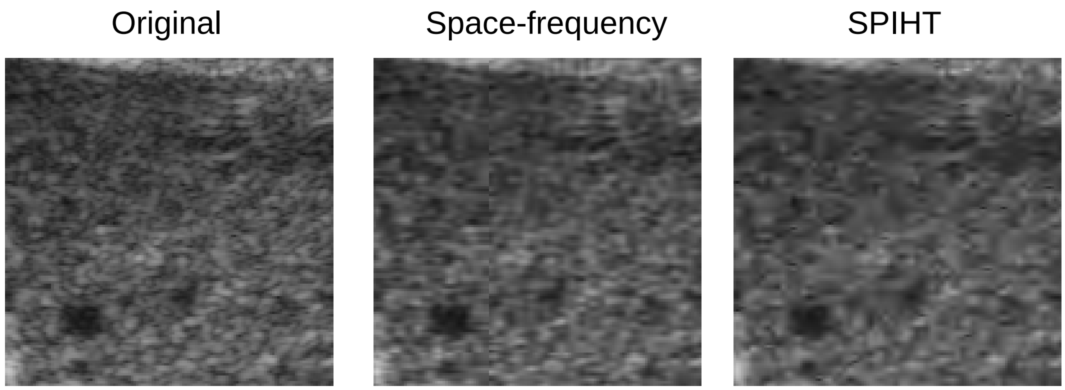


Figure 6.8: Fine details in U1

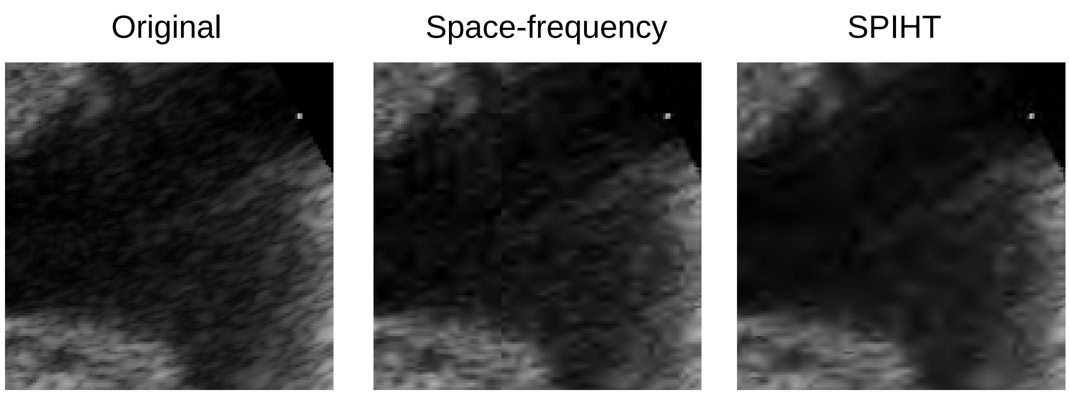


Figure 6.9: Low contrast details in U1

compress. The quality of the space-frequency compressed U7 is quite high, and it is difficult to distinguish it from the original, even in particularly speckly or low contrast areas. The quality of the SPIHT compressed version of U7 is much better than the quality of the SPIHT compressed version of U1, as reflected in the much higher PSNR. Some loss of low contrast detail, for example in the lower left of the image, is still visible in the SPIHT version, as is some blurring in higher contrast fine detail.

Figure 6.10 shows the difference images generated by subtracting the space-frequency estimate from the original for both U1 and U7. The contrast has been increased for

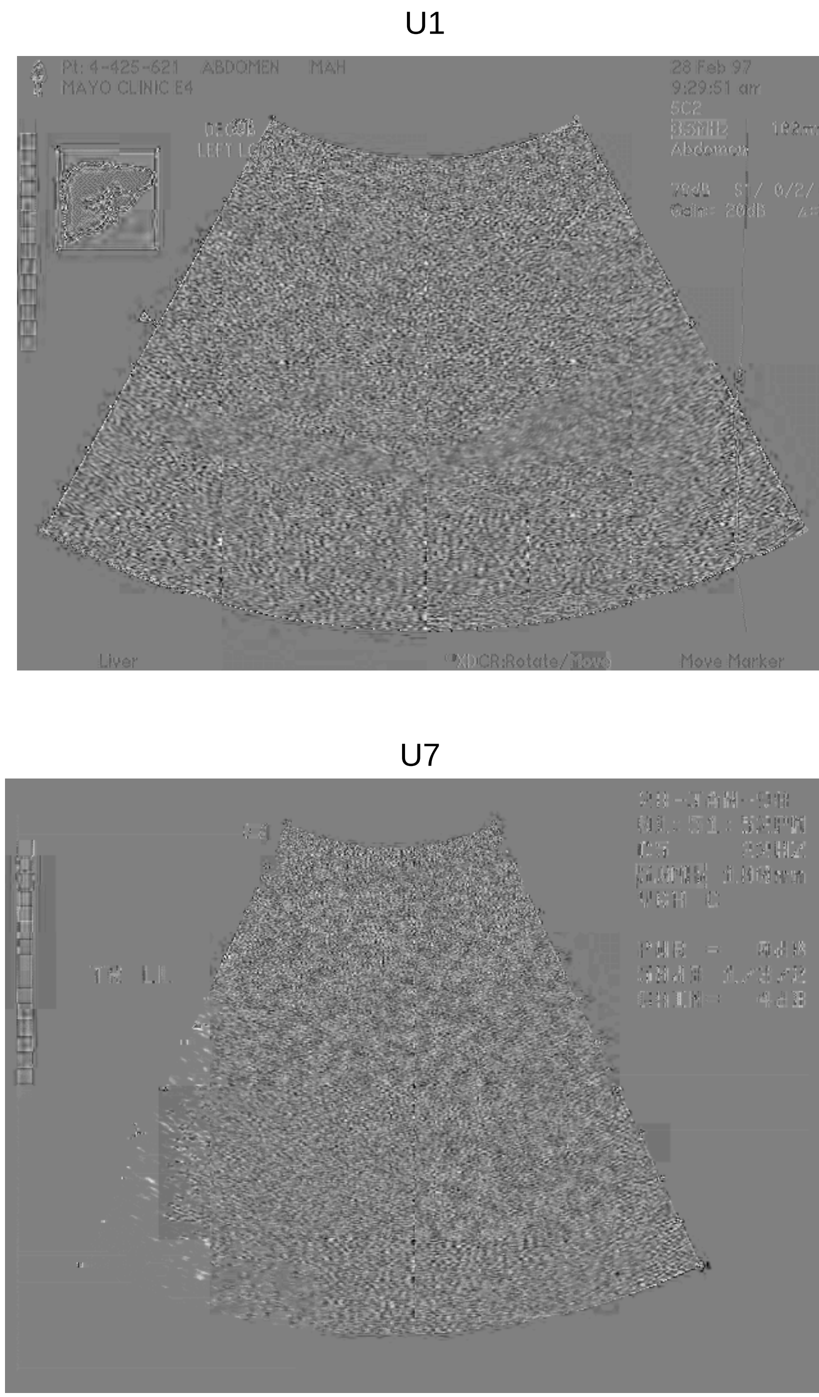


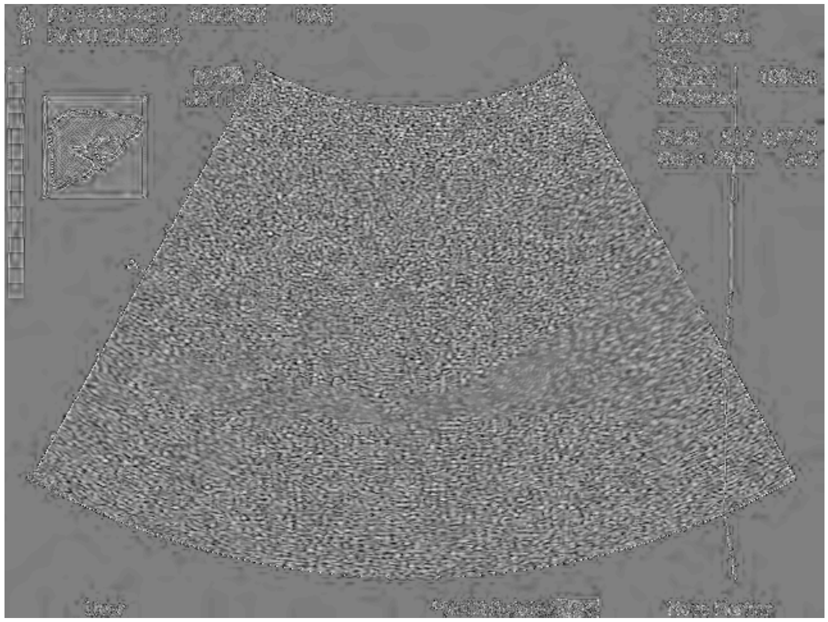
Figure 6.10: Space-frequency difference images

display purposes. (The contrast is increased by a factor of four for the U1 difference image, and by a factor of six for the U7 difference image.) In both cases, the speckle is the most prominent feature of the ultrasound area of the difference image, which indicates that there is a lot of speckle and that it is difficult to code well. For comparison, the difference images generated by subtracting the SPIHT estimate from the original, displayed with the same contrast adjustments as the space-frequency difference images, are shown in Figure 6.11.

The subband images for the space-frequency decomposition of U1 and U7 are shown in Figure 6.12. Recall that a space partition is denoted by a pair of intersecting white lines, and frequency partition is denoted by a pair of intersecting black lines. For both U1 and U7, space partitions dominate the early decomposition levels. Up to depth two, all partitions for both images, with the exception of the depth two frequency partition of the lower right quadrant of U7, are space partitions. The shape of the ultrasound scanned region drives the early partitions, since the optimum representation of the background area and text is to quantize and code it as is, without any frequency decompositions. Since space partitions are limited to binary divisions of the width and height, it is difficult to fit the wedge shaped ultrasound area well. Multiple space partitions early in the decomposition hierarchy limit the depth of subsequent frequency partitions on the ultrasound scanned area, and produce small subbands which limit coding efficiency after only a few frequency partitions.

When the space partitions isolate an ultrasound scanned area, frequency decompositions dominate, and since the depth is limited by the early space partitions, the frequency partitions proceed to the maximum depth in order to accomplish energy compaction. Note that after the first frequency partition, the LH subband is often further partitioned in frequency, but the HL and HH bands are not. As encountered in Section 4.2, the high pass speckle energy is concentrated in the LH band (when the image is oriented such that speckle spots are elongated in the horizontal direction, which is the case here), which explains why additional decomposition of the LH band will produce additional energy

U1



U7

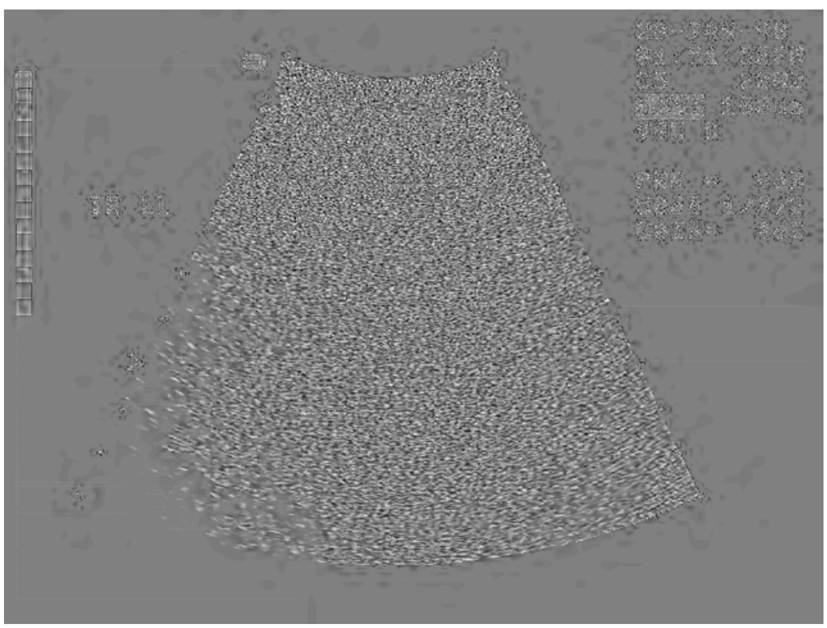


Figure 6.11: SPIHT difference images

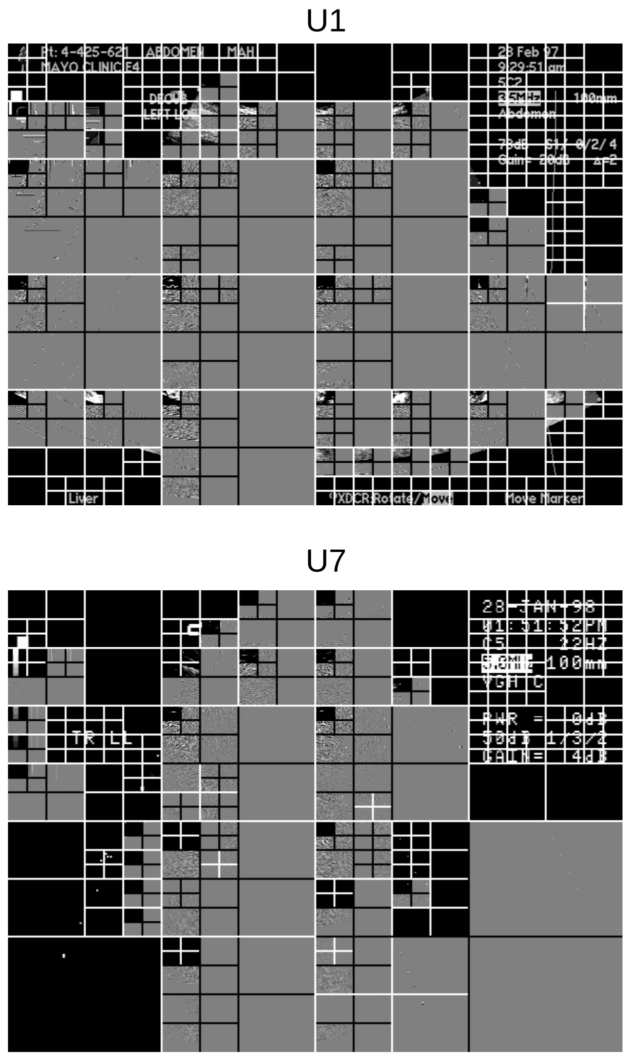


Figure 6.12: Subband images

compaction. This pattern of additional decompositions of the LH band will also appear in the investigation of a common partition for ultrasound only images, which is discussed in Section 6.6.

The optimum decompositions show that the best way to compress ultrasound images is to treat the background and ultrasound scanned areas differently, but this is not a simple task because of the variety of different objects that can appear in the background (text, reverse video text, graphic objects), and the unusual shape of the ultrasound scanned region. Space-frequency segmentation has limitations in dealing efficiently with both background and ultrasound areas simultaneously, but because it finds the optimal representation without making any assumptions about the underlying image, and because it incorporates space partitions that allow it to deal with different regions with different characteristics, space-frequency segmentation performs relatively well for compressing ultrasound images.

6.4.3 Compression experiments: varying codec parameters

To ensure that the assumptions made about codec parameters for the image quality experiments are sane, and to gain some additional insight into the compression of ultrasound images using space-frequency segmentation, the effect of varying one parameter at a time from its baseline value is investigated. Due to the volume of data and the convenience of an objective measure, PSNR is used as the predominant measure of compressed image quality for these experiments.

Target bit rate

The target bit rate is varied from 1.0 bpp down to 0.3 bpp in 0.1 bpp increments. At each 0.1 bpp increment, the space-frequency codec uses a different quantizer set. Using different quantizers for different target rates means coarser quantizers can be used at lower rates, and finer quantizers can be used at higher rates. The same quantizer set is used to code both U1 and U7. The PSNR from the space-frequency codec and from SPIHT is plotted against the rate for U1 and U7 in Figure 6.13. For U1, the space-frequency codec retains its advantage over SPIHT until the lower bit rates, where image quality from both compression schemes is significantly degraded. Since U7 is easier to compress and consequently looks better at the lower bits rates, the narrowing in the space-frequency codec's advantage over SPIHT is less evident for U7.

Number of quantizers

The effect of quantizer set size on the performance of the space-frequency codec at 0.5 bpp is studied by allowing each subband to choose from sixteen, twelve, eight, and

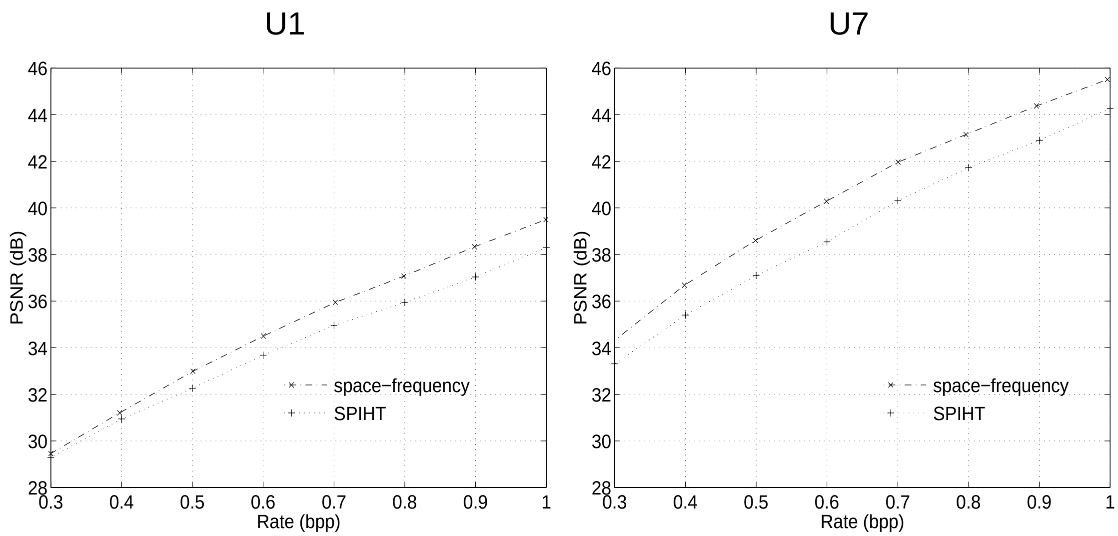


Figure 6.13: Vary the bit rate

four different quantizers. The PSNR is plotted against the number of quantizer choices for U1 and U7 in Figure 6.14. Reducing the number of quantizers reduces the rate by reducing the amount of side information, but reducing the number of quantizers increases the distortion by reducing the choice of available quantizers. The reduction in rate but increase in distortion trade off to produce no significant change in PSNR at a target rate of 0.5 bpp, as the number of quantizers is reduced from sixteen to eight. A noticeable decrease in PSNR does not occur until the number of quantizer choices is reduced to four. Because the cost computation dominates the encoding time, the relationship between the number of quantizer choices and the encoding time is roughly linear. Reducing the number of quantizer choices from sixteen to eight will halve the encoding time, without any noticeable effect on PSNR.

Decomposition depth

The maximum decomposition depth is varied from five to four to three at a target bit rate of 0.5 bpp. The PSNR is plotted against the maximum decomposition depth for U1 and U7 in Figure 6.15. Decreasing the maximum decomposition depth is an effective way of

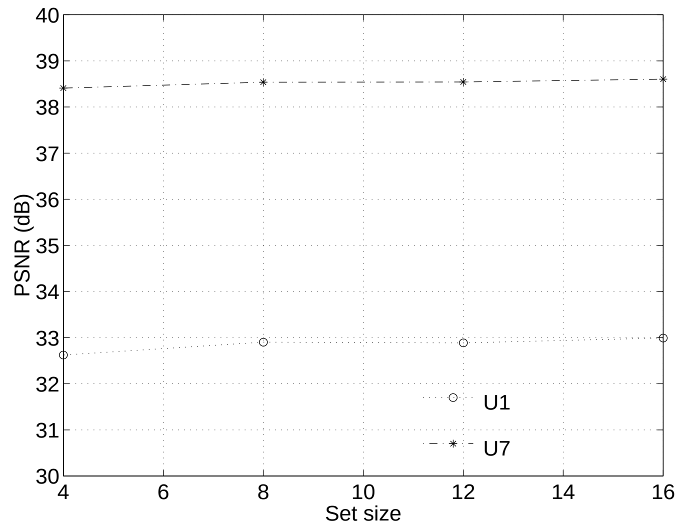


Figure 6.14: Vary the number of quantizers, 0.5 bpp

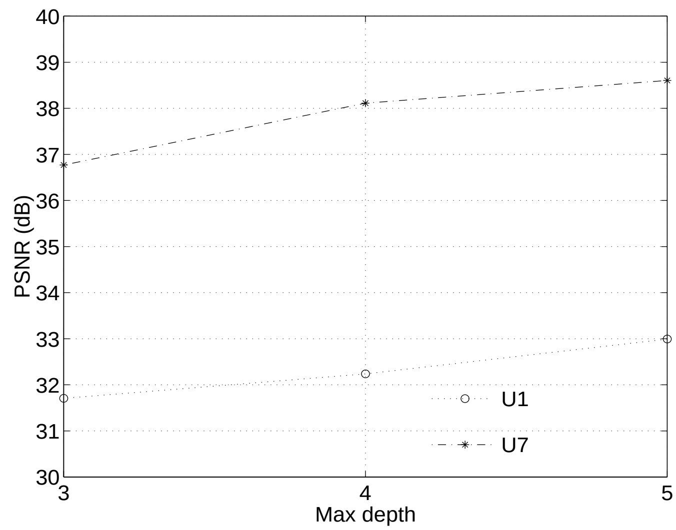


Figure 6.15: Vary the maximum decomposition depth, 0.5 bpp

decreasing the encoding time (decreasing the depth from five to four reduces the encoding time by approximately 40%; decreasing from five to three reduces the encoding time by approximately 60%), however, the increased distortion from reducing the number of space-frequency basis choices means that reducing the depth is not a good way of speeding up the encoding process without compromising image quality.

Entropy coder

In theory, stack-run coding should perform better than arithmetic coding on some sources, and experiments on Lena and Barb indicate that stack-run coding the high pass bands has some advantage over arithmetic coding the high pass bands. In the image quality experiments on U1 and U7, the high pass bands are stack-run coded, but arithmetic coding the high pass bands produces a surprising result. Table 6.5 compares the PSNR on U1 and U7 at 0.5 bpp, where the high pass bands are stack-run coded or arithmetic coded. The low pass and space only bands are arithmetic coded in all cases. Stack-run coding the high pass bands and arithmetic coding the high pass bands produces virtually identical PSNR. For each of the two test images, the optimum partition from the two different coding schemes is quite similar, with predominantly space partitions up to depth two. Subsequent frequency partitions produce small subbands after only a few frequency decomposition stages. Since stack-run coding is not effective at coding small subbands, and the freedom to choose larger subbands while still performing multiple frequency decompositions is limited by the space partitions early in the hierarchy, stack-run coding is not any better than arithmetic coding the high pass bands, for the two ultrasound test images at a target rate of 0.5 bpp.

Table 6.5: Vary the coding method for high pass bands, 0.5 bpp

Image	HP coding method	R (bpp)	PSNR (dB)
U1	Stack-run	0.5008	32.9910
U1	Arithmetic	0.5001	32.9964
U7	Stack-run	0.4991	38.6024
U7	Arithmetic	0.4986	38.6269

Filter set

The choice of filters is investigated by trying two additional biorthogonal filter sets with the space-frequency codec. The additional filters are chosen from those tested by da Silva and Ghanbari [12]. One of the filter sets, labelled 10-18, is chosen in order to use longer filters than the 9-7 Antonini filters, while the other filter set, labelled 6-10, is chosen in order to have filters of similar length to the 9-7 filters. Da Silva and Ghanbari report that both the 10-18 (da Silva and Ghanbari filter “G”) and 6-10 (da Silva and Ghanbari filter “K”) filters perform better than the 9-7 (da Silva and Ghanbari filter “C”) filters in terms of coding gain, PSNR, and subjective grade. In their original paper on stack-run coding, Tsai, Villasenor and Chen [51] use the same 10-18 filter and obtain some improvement in PSNR over the 9-7 filters. The PSNR from compressing the two ultrasound test images at 0.5 bpp using the three filter sets is compared in Table 6.6. The 10-18 filters perform the worst, probably because the block algorithm assumptions are less valid for the longer filters when applied to small subimages. In terms of PSNR, the 6-10 filters perform slightly worse than the 9-7 filters, and in terms of subjective quality, the 6-10 filters seem to preserve slightly less high frequency detail. Of the three filter sets tested, the Antonini 9-7 filters perform the best.

Table 6.6: Vary the filter set, 0.5 bpp

Image	**Filter set**	**R (bpp)**	**PSNR (dB)**
U1	9-7	0.5008	32.9910
U1	10-18	0.4962	32.6211
U1	6-10	0.4956	32.8067
U7	9-7	0.4991	38.6024
U7	10-18	0.4965	38.1496
U7	6-10	0.4964	38.4620

6.4.4 A faster but sub-optimal codec

The main drawback with optimal representation search type codecs is the high computational cost required to find the best representation. The ongoing evolution of microprocessors and computing platforms addresses this problem to a certain extent, but it is useful to find a codec that delivers most of the benefits of the best optimal search while limiting the cost of such a search. The experiments in the previous section provide some clues to the make-up of a faster but sub-optimal space-frequency codec. Arithmetic coding performs as well as stack-run coding on ultrasound images, but zeroth-order entropy can be used to estimate the cost of arithmetic coding. Using zeroth-order entropy in the cost calculation instead of the actual coded rate produces a sub-optimal partition, but also significantly decreases the encoding time, since the cost calculation dominates the encoding process. The number of quantizer choices for each subband can be decreased from sixteen to eight without significant increase in distortion, which approximately halves the encoding time. The maximum decomposition depth is kept at five.

Table 6.7 compares the encoding time and PSNR for the ultrasound test images U1 and U7 at 0.5 bpp, between a faster sub-optimal space-frequency codec and the standard space-frequency codec used in the image quality experiments. The decoding time is

Table 6.7: Fast codec vs. standard codec

Image	**Codec**	**Enc time (min:sec)**	**R (bpp)**	**PSNR (dB)**
U1	Standard	2:15	0.5008	32.9910
U1	Fast	0:17	0.4998	32.7762
U7	Standard	1:52	0.4991	38.6024
U7	Fast	0:16	0.5001	38.4594

1 second in all cases. The speed comparisons are done on a 400 MHz Pentium II Xeon system running Windows NT 4.0. Since computing capabilities improve over time, the absolute encoding time is less significant than the relative reduction in encoding time - 87% in the case of U1 and 86% in the case of U7. The speed improvements are obtained at little cost in terms of PSNR, and in terms of subjective quality the recovered images from the standard codec and the fast codec are very similar.

SPIHT is still much faster than the fast sub-optimal space-frequency codec. The encoding and decoding times for SPIHT are similar. Using SPIHT on the same system, the round-trip (encoding time plus decoding time) coding time required for U1 or U7 is

0.4 seconds. Though the fast space-frequency codec still does not approach the speed of SPIHT, the image quality of the recovered estimates produced by the fast codec is significantly better than SPIHT as measured by subjective quality and PSNR.

6.5 Expert viewer assessment of compressed images

The goal of lossy medical image compression is to reduce the size of the image representation while at the same time to maintain the quality of image features that are relevant for diagnosis. Though they provide a convenient measure of overall image quality, objective measures such as PSNR are usually not designed for medical images. A non-expert viewer may be able to identify compression artifacts, but is not aware of clinically relevant features. Radiologists trained specifically to interpret images from a certain modality are the best judges of medical image quality; this is especially true in the case of diagnostic ultrasound, where the echo location fundamentals behind ultrasound image generation produce images that are quite alien to persons not trained in the interpretation of such images. Images compressed using both the space-frequency codec and SPIHT were presented to ultrasound radiologists to obtain expert viewer assessment of the differences in quality between images from the two different codecs.

In a comprehensive article where they review study methodologies as well as conduct their own study of compressed medical image quality, Cosman, Gray, and Olshen [11] divide the study of perceptual quality into subjective rating of image quality and assessment of diagnostic accuracy. Studies of diagnostic accuracy typically employ receiver operating characteristic (ROC) analysis, where the diagnostic task is defined as a decision with respect to a threshold [30] [10] [35]. ROC studies require definition of a suitable diagnostic task, a large number of test images, availability of some kind of standard diagnosis for each test image, the participation of a reasonable number of radiologists, and sophisticated statistical analysis to interpret the results. Subjective quality assessment is less standardized. In their study of the subjective quality of compressed Computed Radiography chest images, Ishigaki et. al. [30] ask the subjects to rank reconstructed images according to the degree of compression. A more common method of studying subjective quality is to ask the subjects for a quality rating for each compressed image, however the rating scale is not standardized [41] [19].

A valid formal study of compressed medical image quality is clearly a large and demanding task that is beyond the scope of this thesis. The expert viewer assessment task that is attempted is a limited study of subjective quality. The goal of the study is to compare the subjective quality with respect to medically relevant features of ultrasound images compressed with space-frequency segmentation and SPIHT, at a single bit rate which is relevant and interesting.

6.5.1 Setup

The study is conducted with the Radiology Department at Vancouver General Hospital (VGH), and all images used in the expert viewer assessment study are obtained from the Radiology Department at VGH. The images in the test set are chosen with the help of a radiologist to cover a selection that is representative of what radiologists encounter normally, and which includes diagnostically interesting features in the images. Because the VGH Radiology Department conducts a large number of abdominal exams, the images in the test set are dominated by abdominal images, including images of the liver, kidney, gall bladder, and pancreas. Non-abdominal images in the test set consist of two thyroid images. Most of the abdominal images are captured at 4 MHz using harmonic imaging, in which signal processing using the second harmonic produces a slightly sharper image. The rest of the abdominal images are captured at 3.5 MHz. Curved linear array and phased array transducers are used for the abdominal images. The thyroid images are quite different from the other images because they are captured at 8 MHz using linear array transducers, and consequently the ultrasound portion of the thyroid images is rectangular in shape. All images in the test set are grayscale with a resolution of 8 bits. The PSNR of the compressed test images using SPIHT ranges from 35.1 dB to 41.0 dB, while the PSNR using space-frequency segmentation ranges from 37.5 dB to 39.7 dB.

All compressed images in the test set are compressed at a rate of 0.4 bpp, which is slightly lower than the 0.5 bpp used in other experiments in this thesis. The rate of 0.4 bpp is chosen based on non-expert assessment of subjective quality, so that some compression artifacts are visible but overall image quality is still high. The results from the expert viewer assessment will show that this is an appropriate compression rate.

The test methodology is the two alternative forced-choice technique. Subjects are shown a pair of images, and are asked to choose the image with the best subjective quality in terms of medically relevant features. Each pair has a common original image, and no pair of images is exactly identical. The subjects are not told whether or not the images are compressed, nor which compression methods are used on the images. Unlike other subjective tests a quality rating scale is not used, since there is not enough variation in quality at a single bit rate to justify the use of a quality rating, however subjects are encouraged to record any relevant comments about image quality.

Uncompressed images are included in the test set as a control, so that there are three possible permutations for a pair of images: space-frequency vs. SPIHT, uncompressed vs. space-frequency, and uncompressed vs. SPIHT. Twelve pairs of images are used, and since the goal of the test is to compare space-frequency segmentation against SPIHT, six of the twelve pairs are space-frequency vs. SPIHT, while three pairs compare uncompressed vs. space-frequency, and three pairs compare uncompressed vs. SPIHT. Original uncompressed versions of the twelve images are shown in Appendix C. Within the three possible permutations there is a mix of image quality (as measured by PSNR), to minimize the bias towards either space-frequency or SPIHT. To limit learning effects, when the images are presented to the subjects permutations are interleaved in the order: space-frequency vs. SPIHT, uncompressed vs. SPIHT, space-frequency vs. SPIHT, uncompressed vs. space-frequency. Images are presented side-by-side, but the higher quality image in terms of PSNR is randomly distributed to the left or right hand side.

6.5.2 Results

Two radiologists are involved in the study. The study is conducted in the radiology viewing room using monitors that are normally used for ultrasound images, under typical lighting conditions for viewing images. The winners (e.g. the image of the pair with the best quality in terms of medically relevant features) in the twenty four decisions from the two radiologists are tabulated in Table 6.8. Due to unclear instructions to one of the radiologists, there are three “no-decision” results, where the radiologist cannot choose a winner from the pair of images.

Table 6.8: Winners from the study

		Winners
Space-frequency vs. SPIHT	Space-frequency	11
	SPIHT	1
	No decision	0
Uncompressed vs. Space-frequency	Uncompressed	3
	Space-frequency	1
	No decision	2
Uncompressed vs. SPIHT	Uncompressed	5
	SPIHT	0
	No decision	1

The twelve comparisons between compressed images show a clear preference for space-frequency compressed images over SPIHT compressed images. The space-frequency compressed image is chosen as the best quality image of the pair eleven times out of twelve, while the SPIHT compressed image is chosen only once. In the one case where SPIHT is chosen over space-frequency, a thyroid image, the quality of both compressed images is quite high – 39.7 dB for space-frequency and 38.8 dB for SPIHT. The radiologist who chose the SPIHT compressed image preferred it because it appears to have more contrast and sharpness. At high quality levels where objectionable artifacts are minimal, the low pass filtering effect of the wavelet transform can sharpen features by reducing high frequency noise, so that the effects of compression can produce an image that is preferred over the original.

Despite the clear preference for space-frequency compressed images over SPIHT compressed images, at the PSNRs tested, the distinction in quality between the two compression methods with respect to medically relevant features is not large. The radiologist's comments indicate that the winner is often chosen based on better sharpness and higher resolution, particularly in speckly texture regions of an image. In a few cases, the SPIHT compressed image is chosen as the loser because visible artifacts degrade medically significant features. At 0.4 bpp, most of the compressed images in the test set have some loss of detail and exhibit some compression artifacts, but the utility of most of the images for diagnostic purposes is not significantly degraded. When the radiologists focus on compression artifacts, both are always able to identify the image in a pair with lower PSNR, but both radiologists concentrate on using medically significant features as the basis of their winner choice, rather than simply choosing the image with the least visible artifacts. Both radiologists comment that with the exception of a single image, all the images that they are shown are suitable for diagnostic use. The one image that is judged unsuitable is a SPIHT compressed image in which severe artifacts and blurring are visible. The unsuitable image is a low contrast image of a kidney. The bit allocation scheme used in SPIHT, which allocates more bits to larger magnitude coefficients, appears to have trouble with low contrast images, because there is less distinction by magnitude between coefficients.

The decisions from comparisons between compressed and uncompressed images show that for this set of test images, 0.4 bpp is a suitable level of compression for the assessment of compressed image quality. At 0.4 bpp, the quality of space-frequency compressed images is quite high, and it is difficult to choose between space-frequency compressed and uncompressed images, whereas the quality of SPIHT compressed images is degraded, so that the distinction between SPIHT compressed and uncompressed images is generally clear. In the six comparisons between space-frequency compressed and uncompressed images, the space-frequency compressed image is the winner once, the uncompressed image is the winner three times, and there are two "no-decisions". In the six comparisons between SPIHT compressed and uncompressed images, the uncompressed image is the winner five times, and there is one "no-decision". Thyroid images appear easier to compress than abdominal images, and the “no-decision” case for the SPIHT versus uncompressed comparison involves a thyroid image where the PSNR of the SPIHT compressed image is quite high (41.0 dB).

6.6 Partitioning ultrasound-only images

When space-frequency segmentation is applied to medical ultrasound images, where the ultrasound scanned area is often non-rectangular in shape and typically does not occupy

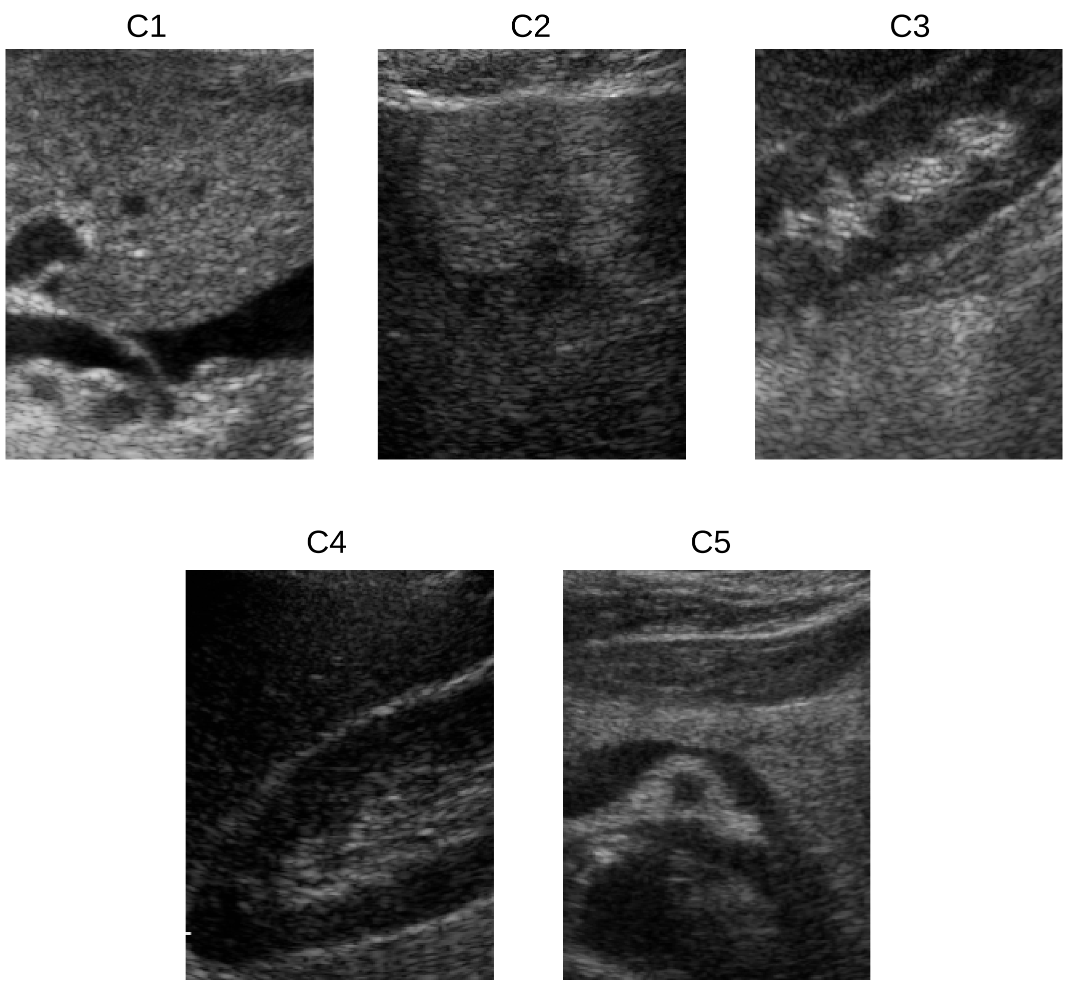


Figure 6.16: Ultrasound-only test images

the entire image, the optimum partition indicates that the background and ultrasound areas of the image should be treated differently by the compression system for best results. Despite the geometric limitations on the possible partitions, space-frequency segmentation does a reasonable job of adapting the optimum partition to fit the shape of the ultrasound region, and does so automatically. If the ultrasound area could be perfectly separated from the background (and ignoring for now exactly how this would be done) and subband coded separately, it is useful to find a good strategy for coding the ultrasound-only area. This leads to the investigation of the optimum partition for ultrasound-only images. In addition, the optimum partitions of ultrasound-only images may give some insight into the characteristics of ultrasound images.

Five rectangular ultrasound-only images are cropped from five medical ultrasound images with wedge shaped ultrasound scanned areas.[4] The wedge shape limits the size of the extracted images to 192×256 pixels. The five images, labelled C1 to C5, are shown in Figure 6.16. The images range in contrast from quite high (C1) to quite dark (C4). All images are quite speckly, and C3, C4, and C5 also contain features from imaged organs.

Space-frequency segmentation is used to code the five ultrasound-only images at 0.5 bpp, using the quantizer set from the image quality experiments in Section 6.4.2. High pass bands are stack-run coded, while low pass and space only bands are arithmetic coded. The filters used are the Antonini 9-7 biorthogonal filters. Because the images are small, the maximum decomposition depth is limited to three. The limited decomposition depth also makes it easier to make sense of the optimum partition for each image. The subband images describing the optimum partition for each image are shown in Figure 6.17.

A common partition is obtained by comparing the decisions from the five optimum partitions at each partition decision point, and choosing the winner by majority vote.The common partition is shown in Figure 6.18. It contains only frequency partitions, and is the same as the optimum partition for C1, C3, and C5. The common partition is quite similar

4. C1 is cropped from U1, and C2 is cropped from U7. The original images used to produce C3-C5 are obtained from the Radiology Department of Vancouver General Hospital.

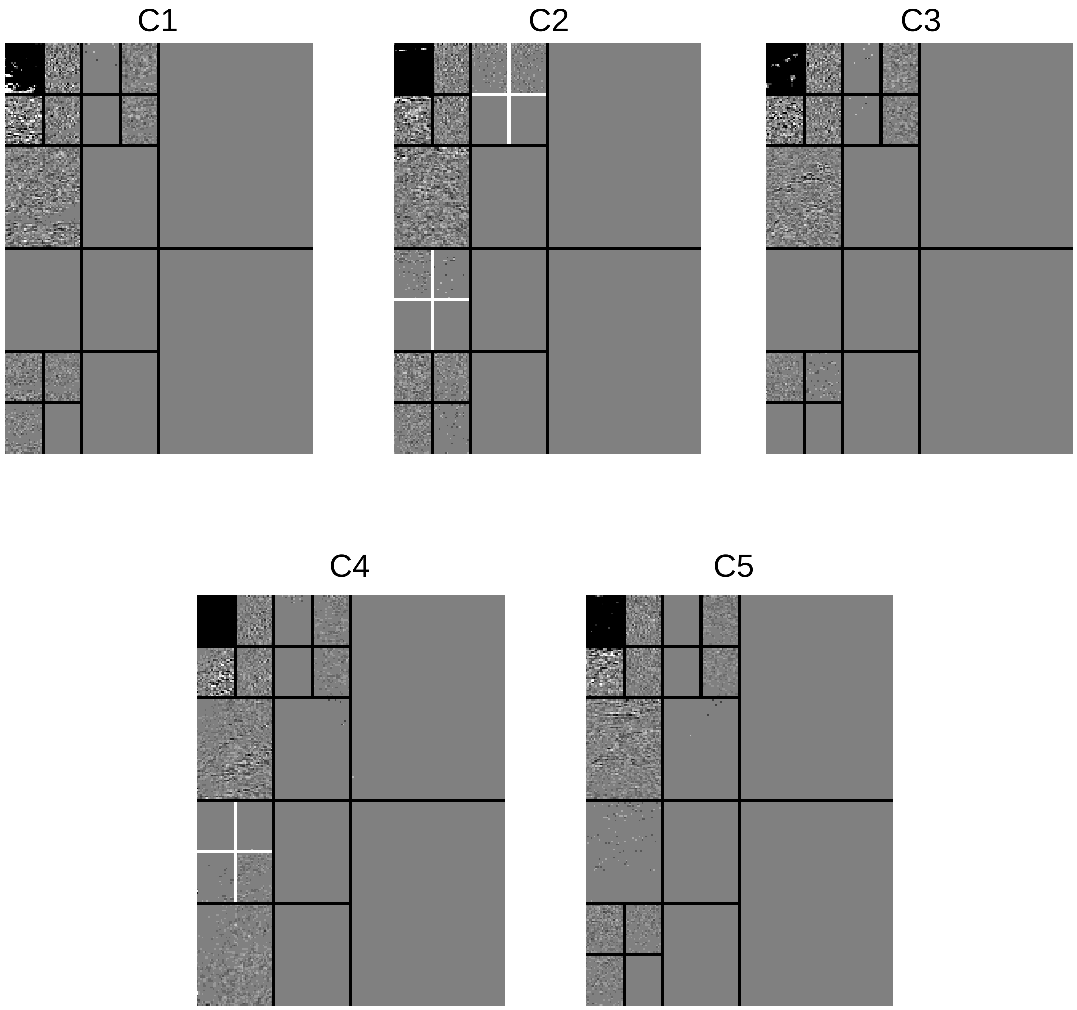


Figure 6.17: Optimum partitions for ultrasound-only test images, 0.5 bpp

to an octave band subband decomposition. Recursive decomposition of the LL band is included in the common partition, and is also part of the optimum partition for all five images, indicating that despite the presence of speckle, ultrasound images are still generally low pass in nature. This is consistent with the statistical analysis of Wagner et. al., which was reviewed in Section 2.5. All five optimum partitions as well as the common partition also include a frequency partition of the LH band at depth one. Partitioning of the

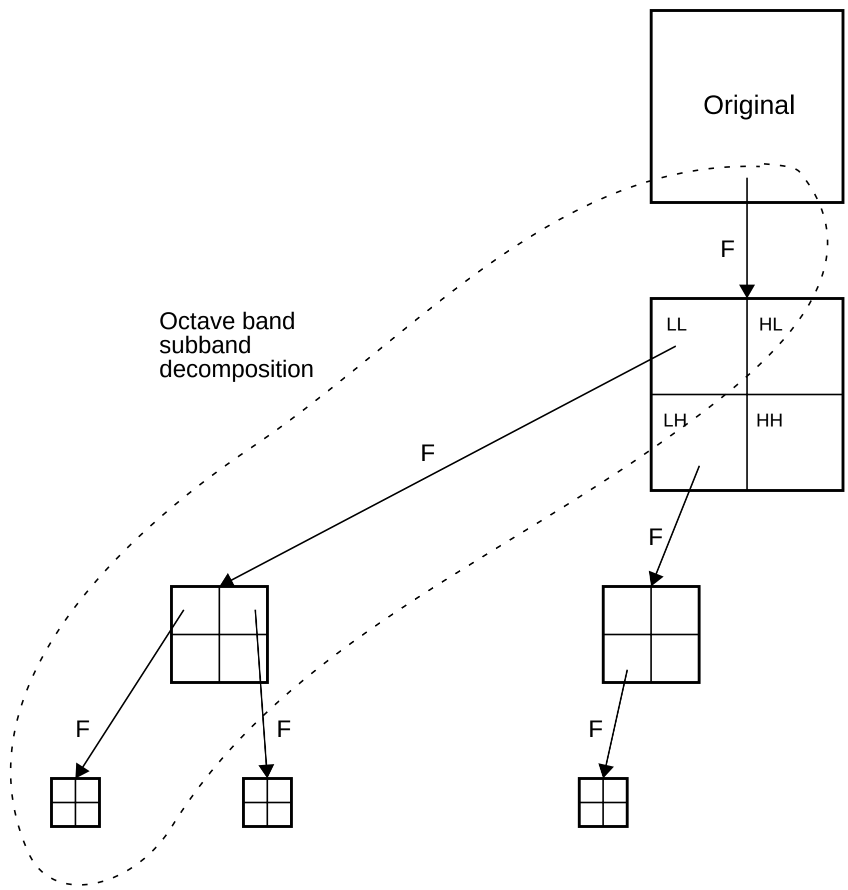


Figure 6.18: The common partition

LH band is consistent with the subband decomposition experiment in Section 4.2, which showed significant energy in the LH band due to the shape of the speckle spots.

The common partition is used to encode the five ultrasound-only images. The PSNR from the common partition encoding is compared against the optimum partition and SPIHT in Table 6.9. The results in the table show that there is little penalty in using the common partition instead of the optimum partition. In all five cases the common partition produces better PSNR than SPIHT. The subjective quality of the estimates recovered from the common partition encoding are superior to the images recovered from SPIHT. The common partition encoding does a better job of preserving high frequency details, particularly the speckle texture. For the ultrasound-only image labelled C1, the original image,

Table 6.9: Common partition vs. SPIHT

Image	Codec	R (bpp)	PSNR (dB)
C1	Optimum partition	0.5164	31.8482
C1	Common partition	0.4953	31.5625
C1	SPIHT	0.500	30.9860
C2	Optimum partition	0.5054	31.8505
C2	Common partition	0.4990	31.7649
C2	SPIHT	0.500	31.3897
C3	Optimum partition	0.5059	33.1806
C3	Common partition	0.4963	32.9311
C3	SPIHT	0.500	32.6403
C4	Optimum partition	0.4992	34.4473
C4	Common partition	0.4961	34.3372
C4	SPIHT	0.500	34.2514
C5	Optimum partition	0.5000	33.6784
C5	Common partition	0.4964	33.6319
C5	SPIHT	0.500	33.4025

the estimate recovered from the common partition codec, and the estimate recovered from SPIHT are displayed side by side in Figure 6.19. The good performance of the common partition with respect to SPIHT indicates that space-frequency segmentation can be a useful tool for finding a good decomposition topology for subband coding ultrasound-only images.

6.7 Summary

Coding experiments on natural images show that the best results are obtained by letting the optimization determine the size of the subimage for coding and quantization. For space-frequency segmentation on natural images, stack-run coding high pass bands works better than arithmetic coding high pass bands. The choice of coding scheme for ultrasound images is less clear, although this may change with higher resolution images that produce larger subbands at a given decomposition depth. On the two test images U1 and U7, stack-run coding and arithmetic coding high pass bands produce virtually identical results, due

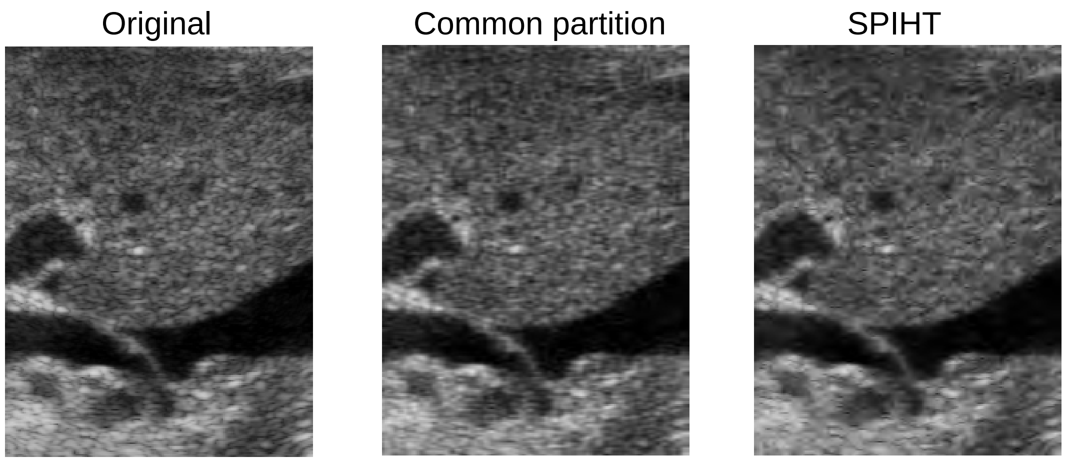


Figure 6.19: C1: original, common partition, and SPIHT, 0.5 bpp

to the early space partitions driven by the shape of the scanned region, which limit the depth of frequency partitions and the size of high pass bands. The similar results using different coding schemes for ultrasound images highlights the power of the rate-distortion optimization algorithm, which exploits the different strengths of different techniques to produce comparable results.

At 0.5 bpp, space-frequency segmentation is better than SPIHT at compressing medical ultrasound images, in terms of both PSNR and subjective quality. Space-frequency segmentation does a better job of preserving both fine detail and low contrast detail. The optimum partition for the space-frequency representation indicates that the best way to compress medical ultrasound images is to treat the background and ultrasound scanned areas differently. Despite its limitations, space-frequency segmentation does a good job at compressing ultrasound images because of its adaptive nature and its ability to treat different regions differently.

A fast sub-optimal space-frequency codec is produced by using entropy as an estimate of rate in the cost calculation, arithmetic coding all subbands, and reducing the number of quantizer choices for each band. Compared to the standard space-frequency codec of Section 6.4.2, the fast codec reduces the encoding time from a few minutes to a few tens of seconds, while maintaining recovered image quality. Though the speed is still significantly lower than that of SPIHT, the quality of the recovered images from the fast space-frequency codec is higher than that of SPIHT in terms of both subjective and objective measures.

Assessment of compressed medical ultrasound image quality by expert viewers shows a clear preference for space-frequency compressed images over SPIHT compressed images. In comparisons between space-frequency compressed and SPIHT compressed images at 0.4 bpp, space-frequency compressed images are consistently chosen as the better quality images in terms of medically relevant features. Despite the consistent distinction between images compressed using the two different methods, the differences in image quality with respect to medically relevant features are not very large.

Experiments on ultrasound-only images show that space-frequency segmentation is useful for finding a good fixed partition for subband coding of ultrasound-only images. A fixed partition is useful for building a fast codec that is optimized for ultrasound images. Experiments using five test images result in a common partition that combined with scalar quantization and stack-run and arithmetic coding performs better than SPIHT in terms of both PSNR and subjective quality.

Chapter 7

Conclusions

Space-frequency segmentation finds the rate-distortion optimal representation of an image, choosing from a set of bases where the possibilities for space or frequency partition are symmetric. The choice of quantizer used to quantize subband coefficients is included in the rate-distortion optimization. Because it adapts the representation to suit the image, space-frequency segmentation is a good choice for images with unusual features. The speckle texture and the shape of the ultrasound scanned region make space-frequency segmentation a good choice for coding medical ultrasound images.

Experiments coding ultrasound images with a space-frequency codec produce good results, showing improvement in terms of both PSNR and subjective quality when compared with wavelet transform zerotree coding using SPIHT. The improvement in subjective quality is particularly noticeable in regions of fine speckle texture. Assessment of compressed ultrasound images by radiologists confirms that, in terms of medically relevant features, the subjective quality of space-frequency compressed images is superior to SPIHT compressed images. The search for an optimal representation is computationally expensive. Using entropy as an estimate of rate in the cost calculation during the optimal search results in a faster codec that produces a close to optimal representation.

7.1 Summary of contributions

A space-frequency segmentation codec is implemented to study the effectiveness of space-frequency segmentation for the compression of medical ultrasound images. The use of actual entropy coders in the implemented codec extends the work of Herley et al., who originally propose the space-frequency algorithm but do not report actual coded rates, by studying the problem of coding the space-frequency representation effectively. The parameters that determine the performance of the space-frequency codec are explored, and a good set of parameter settings is determined experimentally. The use of space-frequency segmentation to compress medical ultrasound images is investigated in detail. An expert viewer study is conducted in which ultrasound radiologists evaluate the subjective quality of compressed medical ultrasound images with respect to diagnostically relevant features.

7.2 Suggestions for future work

There are many parameters that can be varied in the space-frequency codec, such as the choice of quantizers, the entropy coding scheme, and the filters used. Optimization of quantizer set size, step size, alphabet size and granular range requires more experiments to characterize subimage statistics, and should result in improved performance. Choosing different entropy coding schemes based on subimage size is another idea to improve the codec. Since stack-run prefers coding larger subimages, applying arithmetic coding to small subimages and stack-run coding to larger subimages may produce more efficient coding. A similar idea can be used to choose different filters based on subimage size, so that longer filters are applied to larger subimages, but shorter filters are applied to smaller subimages.

Despite improvements in speed, the high computational cost remains the primary drawback in using space-frequency segmentation to code ultrasound images. One area of future research is to improve the speed of the optimal basis search. Because the target images are from a single medical imaging modality and share some common characteristics, searching the entire set of possible space-frequency bases may not be necessary. Investigating ways to limit the search will require a better understanding of the nature of ultrasound images. Calculation of the cost function is the critical path in the space-frequency algorithm. Finding a cost metric that is simple to calculate but results in an efficient basis choice will also improve the speed without sacrificing quality.

Knowledge of the optimal space-frequency representation leads to modifications of fixed transform algorithms to improve their performance on ultrasound images. Treating the background and text separately from the ultrasound scanned area is one obvious method. The utility from treating these areas differently is confirmed by the space partitions in the space-frequency representation, which occur early in the space-frequency decomposition hierarchy. The wavelet transform does not appear to be the best partition choice for preserving fine detail, particularly in speckly regions. Experiments on ultrasound-only images show that space-frequency segmentation can be used as a tool to find an effective fixed partition.

An interesting problem in coding ultrasound images is the need to preserve detail in regions of low contrast. Zerotree methods, which allocate more bits to coefficients with larger magnitude, seem particularly poor at addressing this problem. Space-frequency segmentation appears better at preserving low contrast detail, but because the optimality criterion is minimum overall rate-distortion cost, no particular distinction is made between areas of low or high contrast. A bit allocation scheme which takes into account the local level of contrast is an interesting idea for future investigation.

Bibliography

[1] Antonini, M., M. Barlaud, P. Mathieu, and I. Daubechies, “Image coding using wavelet transform”, *IEEE Transactions on Image Processing,* vol. 1, no. 2, pp. 205-220, April 1992.

[2] Bell, T. C., J. G. Cleary, and I. H. Witten, *Text Compression*. Englewood Cliffs, N.J.: Prentice-Hall, 1990.

[3] Berger, T., *Rate Distortion Theory: A Mathematical Basis for Data Compression*. Englewood Cliffs, N.J.: Prentice-Hall, 1971.

[4] Burckhardt, C. B., “Speckle in ultrasound B-mode scans”, *IEEE Transactions on Sonics and Ultrasonics*, vol. 25, no. 1, pp. 1-6, January 1978.

[5] Cabral, J. E., Jr., D. T. Linker, and Y. Kim, “Compression of pre-scan converted ultrasound sequences”, *Proceedings of SPIE*, vol. 3335, pp. 378-387, 1998.

[6] Chan, Y. T., *Wavelet Basics*. Boston: Kluwer, 1995.

[7] Chou, P. A., T. Lookabaugh, and R. M. Gray, “Optimal pruning with applications to tree-structured source coding and modeling”, *IEEE Transactions on Information Theory*, vol. 35, no. 2, pp. 299-315, March 1989.

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

[9] Coifman, R. R., Y. Meyer, S. Quake, and M. V. Wickerhauser, “Signal processing and compression with wave packets”, Technical Report, Department of Mathematics, Yale University, 1991.

[10] Collins, C. A., D. Lane, M. Frank, M. E. Hardy, D. R. Haynor, D. V. Smith, B. K. Stewart, J. E. S. Parker, G. D. Bender, and Y. Kim, “Design of a receiver operating characteristic (ROC) study of 10:1 image compression”, *Proceedings of SPIE*, vol. 2166, pp. 149-169, 1994.

[11] Cosman, P., R. M. Gray, and R. A. Olshen, “Evaluating quality of compressed medical images: SNR, subjective rating, and diagnostic accuracy”, *Proceedings of the IEEE*, vol. 82, no. 6, pp. 919-932, June 1994.

[12] da Silva, E. A. B. and M. Ghanbari, “On the performance of linear phase wavelet transforms in low bit-rate image coding”, *IEEE Transactions on Image Processing*, vol. 5, no. 5, pp. 689-704, May 1996.

[13] Dainty, J. C., ed., *Laser Speckle and Related Phenomena*. Berlin: Springer-Verlag, 1975.

[14] Daubechies, I., “Orthonormal bases of compactly supported wavelets”, *Communications on Pure and Applied Mathematics*, vol. 41, no. 7, pp. 909-996, October 1988.

[15] Daubechies, I., “The wavelet transform, time-frequency localization and signal analysis”, *IEEE Transactions on Information Theory*, vol. 36, no. 5, pp. 961-1005, September 1990.

[16] Daubechies, I., *Ten Lectures on Wavelets*. Philadelphia: Society for Industrial and Applied Mathematics, 1992.

[17] de Solà Fàbregas, C. and N. P. Tri, “Ultrasound image coding using shape-adaptive DCT and adaptive quantization”, *Proceedings of SPIE*, vol. 3031, pp. 328-338, 1997.

[18] Erickson, B. J., A. Manduca, P. Palisson, K. R. Persons, F. Earnest IV, V. Savcenko, and N. J. Hangiandreou, “Wavelet compression of medical images”, *Radiology*, vol. 206, no. 3, pp. 599-607, March 1998.

[19] Erickson, B. J., A. Manduca, and K. R. Persons, “Clinical evaluation of wavelet compression of digitized chest X-rays”, *Proceedings of SPIE*, vol. 3031, pp. 312-317, 1997.

[20] Farvardin, N. and J. W. Modestino, “Optimum quantizer performance for a class of non-Gaussian memoryless sources”, *IEEE Transactions on Information Theory*, vol. 30, no. 3, pp. 485-497, May 1984.

[21] Fleischer, A. C. and A. E. James Jr., *Introduction to Diagnostic Sonography*. New York: Wiley, 1980.

[22] Fleischer, A. C. and A. E. James Jr., *Diagnostic Sonography: Principals and Clinical Applications*. Philadelphia: W. B. Saunders, 1989.

[23] Gersho, A. and R. M. Gray, *Vector Quantization and Signal Compression*. Boston: Kluwer, 1992.

[24] Goodman, J. W., "Statistical properties of laser speckle patterns", pp. 9-75 in Dainty, J. C., ed., *Laser Speckle and Related Phenomena*. Berlin: Springer-Verlag, 1975.

[25] Gray, R. M., *Source Coding Theory*. Boston: Kluwer, 1990.

[26] Herley, C., J. Kovačević, K. Ramchandran, and M. Vetterli, "Tilings of the time-frequency plane: construction of arbitrary orthogonal bases and fast tiling algorithms", *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3341-3359, December 1991.

[27] Herley, C., Z. Xiong, K. Ramchandran, and M. T. Orchard, "Joint space-frequency segmentation using balanced wavelet packet trees for least-cost image representation", *IEEE Transactions on Image Processing*, vol. 6, no. 9, pp. 1213-1229, September 1997.

[28] Huffman, D. A., "A method for the construction of minimum redundancy codes", *Proceedings of the IRE*, vol. 40, pp. 1098-1101, 1952.

[29] Insana, M. F., R. F. Wagner, B. S. Garra, D. G. Brown, and T. H. Shawker, "Analysis of ultrasound image texture via generalized Rician statistics", *Optical Engineering*, vol. 25, no. 6. pp. 743-748, June 1986.

[30] Ishigaki, T., S. Sakuma, M. Ikeda, Y. Itoh, M. Suzuki, and S. Iwai, "Clinical evaluation of irreversible image compression: analysis of chest imaging with computed radiography", *Radiology*, vol. 175, pp. 739-743, 1990.

[31] Krasner, J., S. B. Lo, B. Garra, and S. K. Mun, "Effect of vector quantization on ultrasound tissue characterization", *Proceedings of SPIE*, vol. 1653, pp. 360-373, 1992.

[32] Luo, J., H. Li, and C. W. Chen, "Ultrasound image coding based on subband decomposition and speckle synthesis", *Proceedings of the International Conference on Image Processing*, vol. 3, pp. 106-109, 1997.

[33] Mallat, S. G., "A theory for multiresolution signal decomposition: the wavelet representation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674-693, July 1989.

[34] McDicken, W. N., *Diagnostic Ultrasonics: Principles and Use of Instruments, 3rd ed*. Edinburgh: Churchill Livingstone, 1991.

[35] McMahon, H., K. Doi, S. Sanada, S. M. Montner, M. L. Giger, C. E. Metz, N. Nakamori, F. Yin, X. Xu, H. Yonekawa, and H. Takeuchi, “Data compression: effect on diagnostic accuracy in digital chest radiography”, *Radiology*, vol. 178, pp. 175-179, 1991.

[36] Oppenheim, A. V. and R. W. Schafer, *Discrete-Time Signal Processing*. Englewood Cliffs, N.J.: Prentice-Hall, 1989.

[37] Pennebaker, W. B. and J. L. Mitchell, *JPEG Still Image Data Compression Standard*. New York: Van Nostrand Reinhold, 1993.

[38] Raffy, P., *Modélisation, Optimisation et Mise en Oeuvre de Quantificateurs bas Débits pour la Compression d'Images Utilisant une Transformée en Ondelettes*. Ph.D. thesis, l'Université de Nice-Sophia Antipolis, December 1997.

[39] Ramchandran, K. and M. Vetterli, “Best wavelet packet bases in a rate-distortion sense”, *IEEE Transactions on Image Processing*, vol. 2, no. 2, pp. 160-174, April 1993.

[40] Rioul, O. and M. Vetterli, “Wavelets and signal processing”, *IEEE Signal Processing Magazine*, vol. 8, no. 4, pp. 14-38, October 1991.

[41] Rowberg, A. H., H. Lee, Y. Kim, M. S. Frank, and H. S. Choi, “Comparison of medical images using prediction and classification”, *Proceedings of SPIE*, vol. 1894, pp. 132-135, 1993.

[42] Said, A. 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*, vol. 6, no. 3, pp. 243-250, June 1996.

[43] Shankar, P. M., “Speckle reduction in ultrasound B-scans using weighted averaging in spatial companding”, *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 33, no. 6, pp. 754-758, November 1986.

[44] Shannon, C. E., “A mathematical theory of communication”, *Bell System Technical Journal*, vol. 27, pp. 379-423, 623-656, 1948.

[45] Shannon, C. E., “Coding theorems for a discrete source with a fidelity criterion”, *IRE National Convention Record*, Part 4, pp. 142-163, 1959.

[46] Shapiro, J. M., "Embedded image coding using zerotrees of wavelet coefficients", *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3445-3462, December 1993.

[47] Shoham, Y. and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 9, pp. 1445-1453, September 1988.

[48] Smith, M. J. T. and T. P. Barnwell III, "Exact reconstruction techniques for tree-structured subband coders", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 3, pp. 434-441, June 1986.

[49] Smith, M. J. T. and S. L. Eddins, "Analysis/synthesis techniques for subband image coding", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 8, pp. 1446-1456, August 1990.

[50] Strang, G. and T. Nguyen, *Wavelets and Filter Banks*. Wellesley, Ma.: Wellesley-Cambridge Press, 1996.

[51] Tsai, M. J., J. D. Villasenor, and F. Chen, "Stack-run image coding", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 5, pp. 519-521, October 1996.

[52] Vaidyanathan, P. P., "Quadrature mirror filter banks, M-band extensions and perfect-reconstruction techniques", *IEEE ASSP Magazine*, vol. 4, no. 3, pp. 4-20, July 1987.

[53] Vaidyanathan, P. P., *Multirate Systems and Filter Banks*. Englewood Cliffs, N.J.: Prentice-Hall, 1993.

[54] Vetterli, M. and C. Herley, "Wavelets and filter banks: theory and design", *IEEE Transactions on Signal Processing*, vol. 40, no. 9, pp. 2207-2232, September 1992.

[55] Vetterli M. and J. Kovačević, *Wavelets and Subband Coding*. Englewood Cliffs, N.J.: Prentice-Hall, 1995.

[56] Wagner, R. F., M. F. Insana, and D. G. Brown, "Unified approach to the detection and classification of speckle texture in diagnostic ultrasound", *Optical Engineering*, vol. 25, no. 6, pp. 738-742, June 1986.

[57] Wagner, R. F., S. W. Smith, J. M. Sandrik, and H. Lopez, "Statistics of speckle in ultrasound B-Scans", *IEEE Transactions on Sonics and Ultrasonics*, vol. 30, no. 3, pp. 156-163, May 1983.

[58] Woods, J. W., ed., *Subband Image Coding*. Boston: Kluwer, 1991.

[59] Woods, J. W. and S. D. O'Neil, "Subband coding of images", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 5, pp. 1278-1288, October 1986.

[60] Zeng, X., E. A. Geiser, A. F. Laine, and D. C. Wilson, "Homomorphic wavelet shrinkage and feature emphasis for speckle reduction and enhancement of echocardiographic images", *Proceedings of SPIE*, vol. 2710, pp. 658-667, 1996.

Appendix A

Quantizer Set for Lena and Barb, 0.5 bpp

Table A.1: Quantizers for high pass bands

Step size	**Alphabet size**	**Minimum output**
128.0	64	-4096.0
84.0	64	-2688.0
78.9	64	-2524.8
73.7	64	-2358.4
68.6	64	-2195.2
63.4	64	-2028.8
58.3	64	-1865.6
53.1	64	-1699.2
48.0	64	-1536.0
42.9	64	-1372.8
37.7	64	-1206.4
32.6	64	-1043.2
27.4	64	-876.8
22.3	64	-713.6
17.1	64	-547.2
12.0	64	-384.0

Table A.2: Quantizers for space-only bands

Step size	Alphabet size	Minimum output
128.0	3	-128.0
120.0	4	-128.0
112.0	4	-128.0
104.0	4	-128.0
96.0	4	-128.0
88.0	4	-128.0
80.0	5	-128.0
72.0	5	-128.0
64.0	5	-128.0
56.0	6	-128.0
48.0	7	-128.0
40.0	8	-128.0
32.0	9	-128.0
24.0	12	-128.0
16.0	17	-128.0
8.0	33	-128.0

Table A.3: Quantizers for low pass bands

Step size	Alphabet size	Minimum output
64.0	64	-2048.0
61.3	64	-1961.6
58.7	64	-1878.4
56.0	64	-1792.0
53.3	64	-1705.6
50.7	64	-1622.4
48.0	64	-1536.0
45.3	64	-1449.6
42.7	64	-1366.4
40.0	64	-1280.0
37.3	64	-1193.6
34.7	64	-1110.4
32.0	64	-1024.0
29.3	64	-937.6

Table A.3: Quantizers for low pass bands

Step size	**Alphabet size**	**Minimum output**
26.7	64	-854.4
24.0	64	-768.0

Appendix B

Quantizer Set for U1 and U7, 0.5 bpp

Table B.1: Quantizers for high pass bands

Step size	Alphabet size	Minimum output
128.0	64	-4096.0
64.0	64	-2048.0
60.0	64	-1920.0
56.0	64	-1792.0
52.0	64	-1664.0
48.0	64	-1536.0
44.0	64	-1408.0
40.0	64	-1280.0
36.0	64	-1152.0
32.0	64	-1024.0
28.0	64	-896.0
24.0	64	-768.0
20.0	64	-640.0
16.0	64	-512.0
12.0	64	-384.0
8.0	64	-256.0

Table B.2: Quantizers for space-only bands

Step size	**Alphabet size**	**Minimum output**
128.0	3	-128.0
44.0	7	-128.0
41.7	8	-128.0
39.4	8	-128.0
37.1	8	-128.0
34.9	9	-128.0
32.6	9	-128.0
30.3	10	-128.0
28.0	11	-128.0
25.7	11	-128.0
23.4	12	-128.0
21.1	14	-128.0
18.9	15	-128.0
16.6	17	-128.0
14.3	19	-128.0
12.0	23	-128.0

Table B.3: Quantizers for low pass bands

Step size	**Alphabet size**	**Minimum output**
128.0	64	-4096.0
40.0	64	-1280.0
38.0	64	-1216.0
36.0	64	-1152.0
34.0	64	-1088.0
32.0	64	-1024.0
30.0	64	-960.0
28.0	64	-896.0
26.0	64	-832.0
24.0	64	-768.0
22.0	64	-704.0
20.0	64	-640.0
18.0	64	-576.0
16.0	64	-512.0

Table B.3: Quantizers for low pass bands

Step size	**Alphabet size**	**Minimum output**
14.0	64	-448.0
12.0	64	-384.0

Appendix C

Images for the Expert Viewer Study

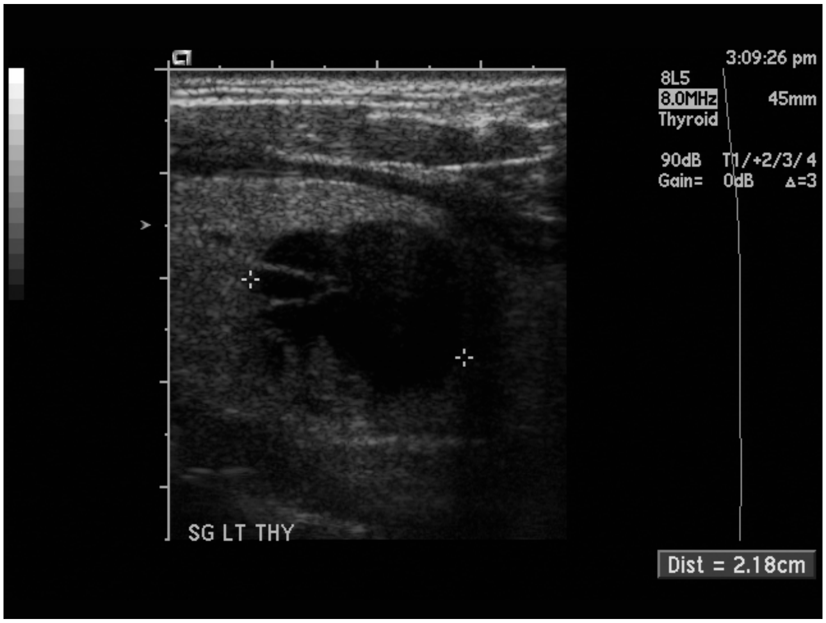


Figure C.1: Space-frequency vs. SPIHT, Image 1

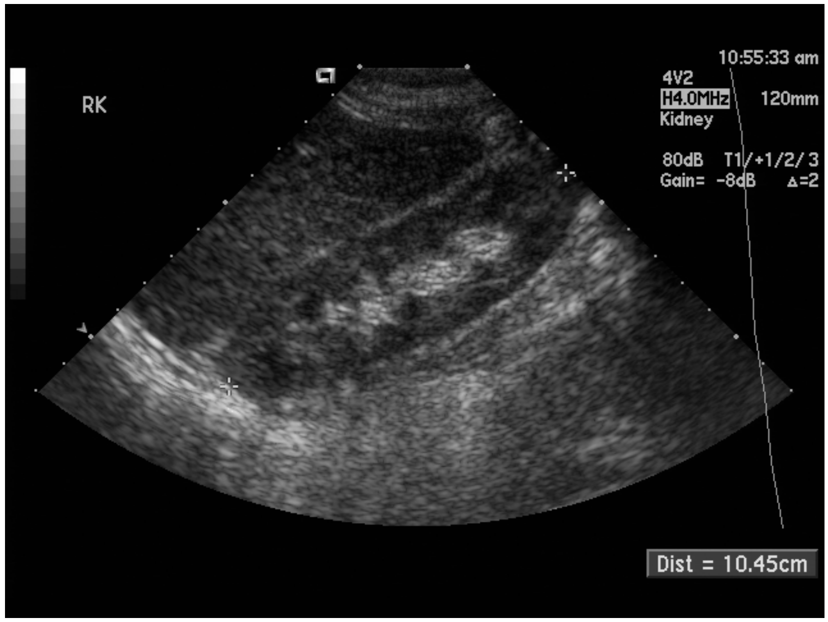


Figure C.2: SPIHT vs. original, Image 1

Figure C.3: Space-frequency vs. SPIHT, Image 2

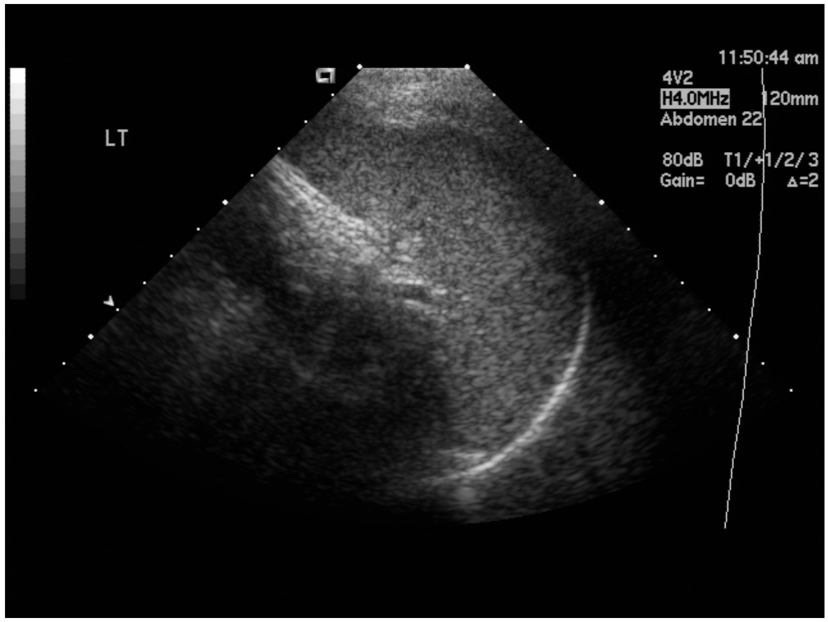


Figure C.4: Space-frequency vs.original, Image 1

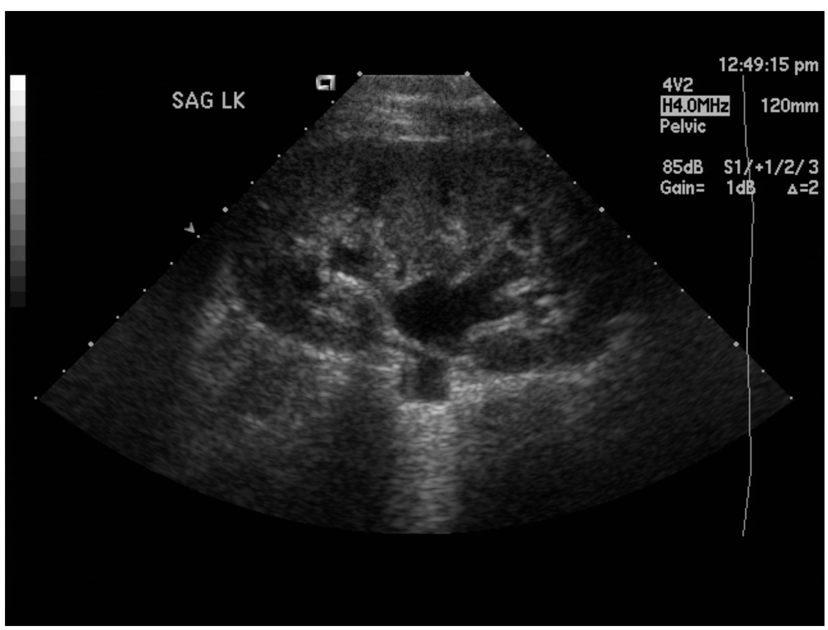


Figure C.5: Space-frequency vs. SPIHT, Image 3

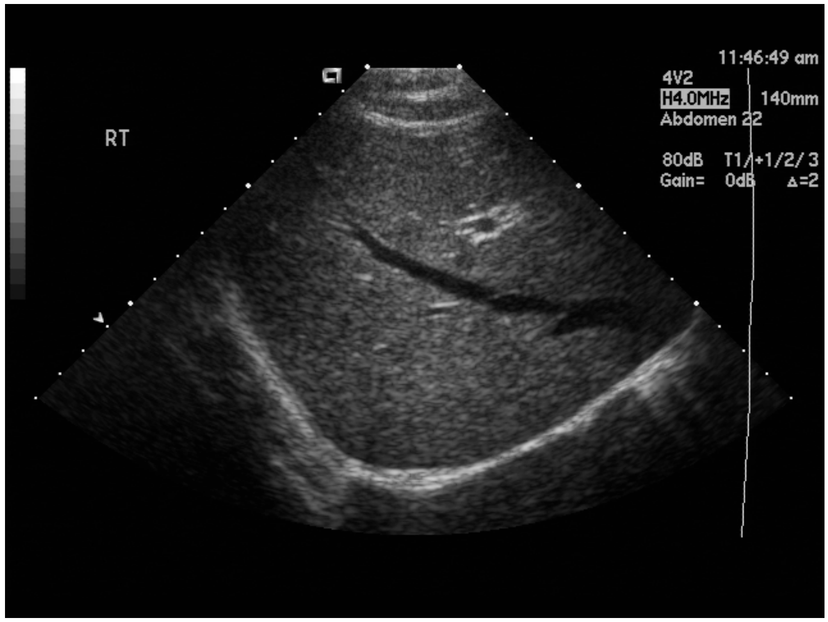


Figure C.6: SPIHT vs. original, Image 2

Figure C.7: Space-frequency vs. SPIHT, Image 4

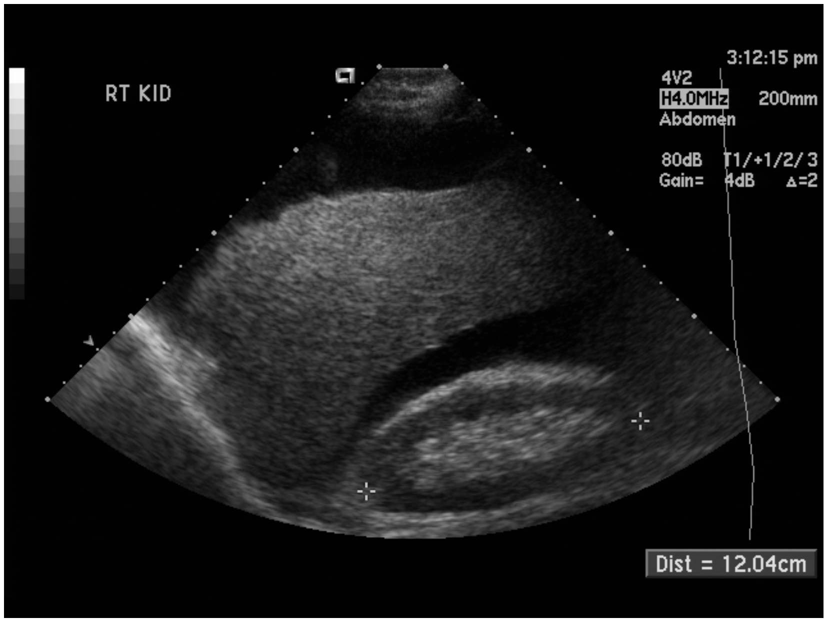


Figure C.8: Space-frequency vs. original, Image 2

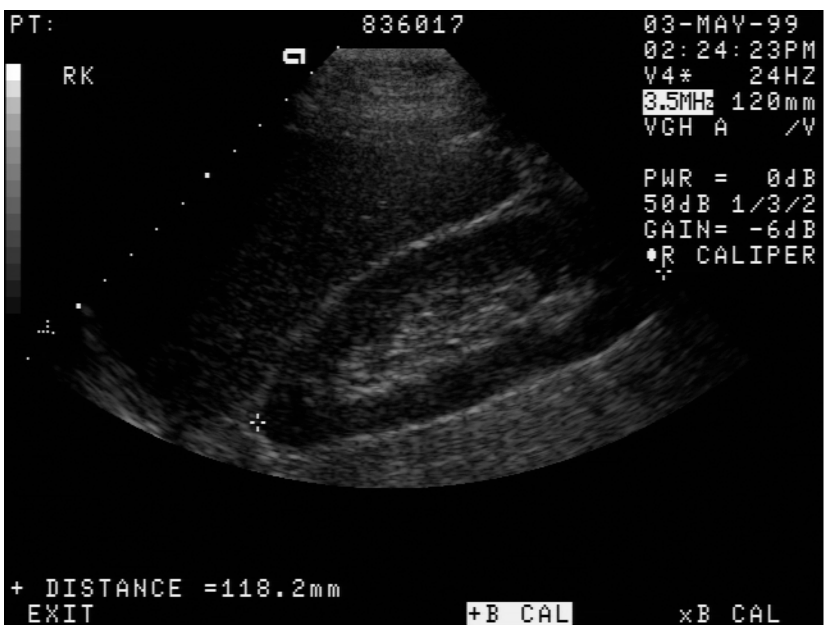


Figure C.9: Space-frequency vs. SPIHT, Image 5

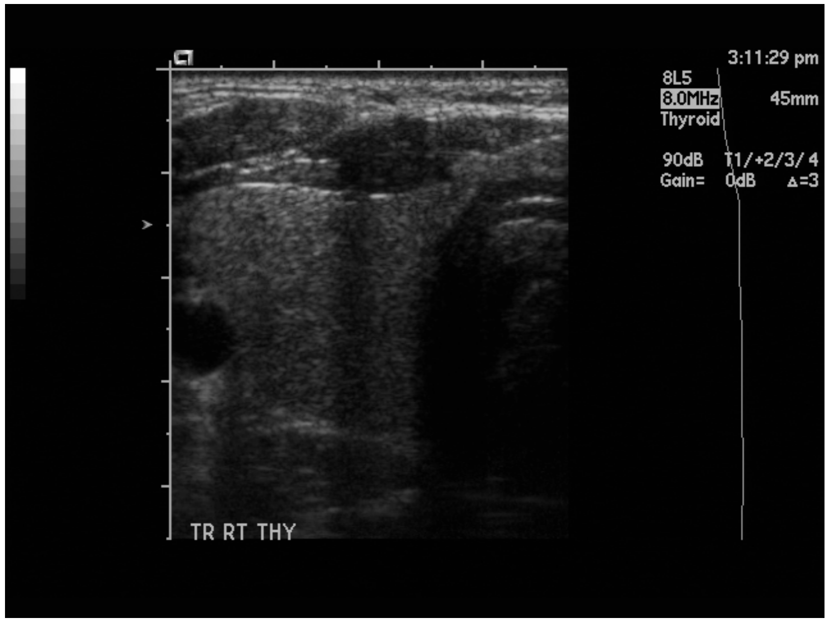


Figure C.10: SPIHT vs. original, Image 3

Figure C.11: Space-frequency vs. SPIHT, Image 6

Figure C.12: Space-frequency vs. original, Image 3