CONTEXT-BASED ENTROPY CODING WITH SPACE-FREQUENCY SEGMENTATION IN ULTRASOUND IMAGE COMPRESSION

by

Chen Ji

B.A.Sc., Shanghai Jiao Tong University, 1990

M.A.Sc., Shanghai Jiao Tong University, 1993

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

May 2001

Approval

Name: Chen Ji

Degree: Master of Applied Science

Title of thesis: Context Based Entropy Coding with Space-Frequency Segmentation in Ultrasound Image Compression

Examining Committee:

Chair: Dr. James K. Cavers

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

Entropy coding provides for the lossless compression of data symbols and is a critical component in signal compression algorithms. In the case where there is statistical dependence between the symbols produced by the source, the performance of an entropy coder can be significantly improved by having the coder dynamically adapt to the current "context".

This thesis considers the specific case of the Space-Frequency Segmentation (SFS) compression algorithm, a method that has been successfully used in the compression of ultrasound images, but which assumes essentially independent symbols in the design of its entropy coder. In particular, we focus on different approaches to picking the context set to use for the entropy coder, as well as the tradeoffs between the number of different contexts to use and the difficulty in reliably estimating the symbol statistics. These issues are addressed for both natural and ultrasound images to determine the effect of the image type.

Experiments on natural and ultrasound images show that our modifications to the basic SFS algorithm, results in significantly better image quality at comparable compression rates. Our method also performs well in comparison to commonly accepted image coding benchmarks, such as the Set Partitioning in Hierarchical Trees (SPIHT) algorithm.

Acknowledgments

I would like to thank my senior supervisor, Dr. Jacques Vaisey, for his invaluable guidance and support throughout the entire process of this thesis research work. Special thanks to my supervisor, Dr. Stella Atkins, who has given me important suggestions during my research. I am also grateful to Dr. Ze-Nian Li for taking time to act as thesis examiner and to Dr. James K. Cavers for chairing the examining committee.

I am grateful to Ed Chiu who provides the software of Space-Frequency Segmentation algorithm and some ultrasound images. Thanks also to Subbalakshmi, Florina Rogers, Tong Jin, Jerry Zhang and Ming Li, for making our lab friendly and lively.

Contents

List of Tables

List of Figures

List of Abbreviations

bpp	bit per pixel
CALIC	Context-based Adaptive Lossless Image Coding
CCBEC	Causal Context Based Entropy Coder
CFH	Cumulative Frequency Histogram
CREW	Compression with Reversible Embedded Wavelets
CSQM	Context Selection, Quantization and Modeling
DPCM	Delta Pulse Code Modulation
DWT	Discrete Wavelet Transform
EBCOT	Embedded Block Coding with Optimized Truncation of the embedded bit-streams.
ECECOW	Embedded Conditional Entropy Coding Of Wavelet coefficients
EPWIC	Embedded Predictive Wavelet Image Coder
EZW	Embedded image coding using Zerotrees of Wavelet coefficients
GAP	Gradient Adjusted Prediction
HINT	Hierarchical INTerpolation
HIS	Hospital Information System
ISO	International Standards Organization
ITU-T	International Telecommunication Union – Telecommunication standardization sector
JPEG	Joint Photographers Expert Group
LZC	Layered Zero Coding
MED	Median Edge Detection
MPEG	Moving Picture Experts Group

MSE	Mean Square Error
NCBEC	Non-causal Context Based Entropy Coder
PACS	Picture Archiving and Communication Systems
PCM	Pulse Code Modulation
pdf	probability density function
PSNR	Peak Signal to Noise Ratio
RIS	Radiology Information System
SFQ	Space-Frequency Quantization for Wavelet Image Coding
SFS	joint Space-Frequency Segmentation using balanced wavelet packet trees for least-cost image representation
SPIHT	Set Partitioning In Hierarchical Trees
SR	Stack-Run image coding
TIDE	Textual Information Detection and Elimination
WP	Wavelet Packets

List of Symbols

$H(S)$	entropy of source S
$p(s_j)$	probability of symbol s_j
s^2 or s_d^2	mean square error
$\mathbf{L}_0$	low pass analysis filter
$\mathbf{H}_0$	high pass analysis filter
$\mathbf{L}_1$	low pass synthesis filter
$\mathbf{H}_1$	high pass analysis filter
$y(t)$	the wavelet
$j(t)$	the scaling function
$J(l)$	Lagrangian cost function
l	Lagrangian multiplier
$D(x)$	the distortion at x
$R(x)$	the rate at x
d_h	the horizontal gradient
d_v	the vertical gradient
$m[i, j]$	the sub-sampled image
t	the spatial texture patterns
Δ	the error strength discriminant
Q_N	quantizer N
L_l	quantization layer l
$k_n[i, j]$	conditioning term
$k'_n[i, j]$	local scale conditioning information
$\oplus$	direct summary
i	logical or
r_fix	rate with fixed context size

r_opt	rate with optimized context size

Chapter 1

Introduction

Entropy coding techniques are important components in image coding algorithms. They perform the lossless compression on symbols to obtain a more efficient representation of source data. In the case where the source data possesses certain statistical dependencies between symbols, it is often advantageous for an entropy coder to consider this statistical property in the source data. Furthermore, if an entropy coder can dynamically adapt to the statistics of the input symbols, better performance will generally be achieved than with its non-adaptive counterpart. In this thesis, ultrasound images compression is used as the platform for the development of context-based entropy coding algorithms in conjunction with space-frequency segmentation.

Diagnostic sonography is usually referred to as ultrasound. It is widely accepted as a popular clinical imaging modality in medical practices due to its fundamental virtues of being noninvasive, allowing real time imaging, and offering no X-ray radiation hazards to either the patient or the sonographer. Huge volumes of data for ultrasound images have been generated as a result of its widespread applications. Despite the availability of inexpensive hardware that can store immense amounts of data, image compression is still desirable for digital storage because of the need for efficient communication and efficient storage of medical images, especially with the rapid expansion of computer networked consultation.

Because ultrasound images are used for medical diagnosis, any visual artifact in the reconstructed images after compression may hinder diagnostic conclusions and lead to severe consequences. Therefore, preserving diagnostically relevant information and maintaining acceptable aesthetic quality is critical in ultrasound image coding. This goal is often achieved by sacrificing compression efficiency, leading to an engineering compromise. The advancement in ultrasound image compression proposed in this dissertation relies upon efficient lossy compression to allow for the preservation of high quality images.

1.1 Motivation

Ultrasound imaging is a technique that uses reflected ultrasonic waves to display visual images of structures within the body. The images generated are stored as digital data, and accessed and transmitted as archival records of physical exams for diagnostic and surgical usage in hospitals or health care centers. A typical eight bit grayscale ultrasound image has 640x480 pixels, which means 2.46 megabits (Mb) of data. In a single medical exam there are on average 10 to 20 still ultrasound images generated equaling to approximately 24.6 to 49.2Mb of data. The data in a clip of ultrasound video, which is becoming popular and practical in hospitals, requires substantially more space for storage than still images. With rising numbers of patients and elongated case histories, ultrasound images/videos accumulate, rapidly filling limited storage space. An additional factor is the recent development of medical Picture Archiving and Communications Systems (PACS), and Hospital Information System (HIS) and Radiology Information System (RIS). PACS/HIS/RIS systems produce electronic images as electronic patient records, which can be integrated within a central information system. The connected system provides a sophisticated means for processing medical information from various sources, which results in a high demand for transmitting and storing enormous amount of data. A well designed data compression algorithm reduces the file size and makes storage, access, and transmission practical and efficient.

Another incentive for efficient ultrasound image compression is telemedicine, which is the electronic transmission of medical information between different health care sites. Telemedicine may require the transmission of large medical images containing diagnostic information.

Overall, digital storage constraints and tele-clinical practice requirements are two important issues for ultrasound image compression. Furthermore, diagnostic usage requires that the relevant medical information contained within the image be well preserved after compression. These two requirements on ultrasound image compression techniques motivate the research produced by this thesis, which endeavors to find an efficient lossy compression algorithm to achieve both a high quantitative fidelity as determined by the Peak Signal-to-Noise Ratio (PSNR), and a high qualitative image quality for human examiner.

Among recent image compression techniques, wavelet-based methods are considered to be among the most effective [9]. The ongoing

JPEG2000, the new ISO/ITU-T standard for still image coding, is based on Discrete Wavelet Transform (DWT), scalar quantization, context modeling, arithmetic coding and post-compression rate allocation [34]. From a functionality point of view, JPEG2000 provides superior performance to a lot of algorithms including the current JPEG standard created in the late 1980s, MPEG-4 Visual Texture coding (VTC), JPEG-LS which is the latest ISO/ITU-T standard for lossless coding of still images, and the Portable Network Graphics (PNG) algorithm [34]. However, for a particular application, the choice depends on the requirements. In the cases where lossy compression is of high priority and low complexity is of interest, SPIHT provides a good solution in recent times [4]. Both JPEG2000 and SPIHT are famous image compression algorithms using wavelet transform to generate the subband samples which are to be quantized and coded. SPIHT [4] uses the dyadic decomposition structure, and JPEG2000 applies not only the dyadic decomposition structure, but also other “packet” partitions [20]. These two algorithms are described in detail in Section 2.1.3. Besides JPEG2000 and SPIHT, another wavelet-based lossy image coding scheme is the joint Space-Frequency Segmentation (SFS) developed by Herley et al.[13]. The SFS algorithm uses balanced wavelet packet trees to decompose the image into different subbands, reduce the redundant information and allow subbands to be coded separately based on their different statistics. The wavelet transform used in this algorithm concentrates energy in the low pass bands, and is effective for coding natural images, since they generally have little high frequency content.

Ultrasound images have some unique features that make them different from natural images. For example, the “speckle”, an ultrasonic scanning artifact caused by scattered reflections, concentrates its energy in certain spectral regions and gives rise to increased energy in certain high pass subbands [22]. Herley's SFS algorithm was tested on ultrasound images [22] and obtained superior results over other wavelet-based coding schemes, such as SPIHT, from results of both subjective assessment by radiologists and objective evaluation of PSNR values. This adaptive topology has shown its potential on ultrasound images. Therefore, it is chosen as the segmentation scheme for the proposed coding algorithm in this thesis. However, the entropy coders considered in this scheme are either the classical arithmetic coder or the stack run coder [22] that take little advantage of the intra-subband dependencies. The insufficient exploitation of these dependencies leads to sub-optimal choices for space-frequency partition; as a result, a higher order entropy coder is expected to improve the compression performance.

1.2 Summary of Contributions

The main contributions of this thesis include developing and applying an efficient context-based entropy coding algorithm that can be used on SFS segmented subbands to improve compression performance, particularly in the case of ultrasound image compression. The proposed algorithm focuses on an innovative scheme for adaptive context-based entropy coding technique and its application on the quantized SFS subband coefficients. Experiments conducted using the new algorithm produced encouraging results.

The higher ordered entropy coder using our proposed context model is more efficient than the basic arithmetic coder. It provides more appropriate rate-distortion optimization for the space-frequency segmentation than the basic arithmetic coder does. Therefore, when the original SFS algorithm is combined with the proposed context-based entropy coder, higher PSNR values at various target bit rates are obtained. The improvement of the compression performance shows the potential of the context-based entropy coder on lossy image compression.

1.3 Thesis Outline

The remaining of this thesis is structured as follows:

Chapter 2 provides background information on digital image compression techniques and basic ultrasound image analysis, upon which the rest of the thesis is founded. Previous research on efficient ultrasound image compression algorithms is surveyed in Chapter 3. Chapter 4 compares various context-based entropy codecs that use different context models and presents explanations on why these models are chosen and how they work. Experimental evaluations, such as comparisons between context models, and a combination of context-based entropy coder and space-frequency segmentation are reported in Chapter 5. An analysis on subband coefficients statistics for histogram initialization is given in Chapter 6. Finally, we provide some conclusions and suggestions for future work in Chapter 7.

Chapter 2
Background

Image coding is an essential component in digital data communication and storage systems. Technically speaking, coding attempts to minimize the number of information-carrying units used to represent an image [6]. In this chapter, the fundamentals of digital image compression techniques are introduced and basic coding schemes are reviewed. Ultrasound image characteristics are also discussed.

2.1 Digital Image Compression

Digital image compression, also known as digital image coding, or source coding, is a technique that reduces the amount of data used to represent source information, therefore minimizing the necessary storage space or transmission time for a given channel capacity. The goal of image compression is to obtain a more efficient representation of source data while preserving the essential information. In recent years, techniques for removing redundancy from highly-correlated images have been extensively studied. Some popular techniques addressed in the literature include vector quantization (VQ), linear predictive coding, linear transform coding (like the KLT and the DCT), and subband coding as well as combinations of them. The following sections introduce the following basic techniques:

arithmetic coding, subband coding, wavelet transforms, and context-based compression methods.

2.1.1 Introduction to Image Coding Techniques

Entropy, introduced by Claude Shannon [49], is a concept of measuring the uncertainty of source information. Mathematically speaking [55], suppose there is a zeroth-order memoryless statistics source modeled independently as a discrete random process with certain probabilities. Given the symbol set S, where $S=\{s_j \mid j = 0,1,\ldots M-1\}$, if the source produces a sequence of random variables, $r_0, r_1, r_2\ldots$ where $r_k \in S$, having the probability model $P = \{p(s_j) \mid j = 0,\ldots,M-1; s_j \in S\}$, the information of the source is then defined as its entropy[1] :

$$H(S) = -\sum_{j=0}^{M-1} p(s_j)\log p(s_j) \qquad 2.1$$

Shannon [49] showed that the best that a lossless compression scheme can do is to encode a memoryless source with an average number of bits equal to the entropy of the source defined above. The zeroth-order entropy does not utilize memory among the source data. So the above theory only works for the case of memoryless sources. When memory is considered while encoding, dependencies between the symbols will be exploited by adapting the coder to the current “context” and higher compression ratio will be achieved through the context-based entropy coder. Therefore a higher-ordered entropy coder that considers memory among symbols can encode the source with bits less than its zeroth-order entropy.

---

[1] Strictly speaking, it should be called zeroth-order entropy. Sometimes [25], it is also referred as the first-order entropy.

Generally speaking, image coding techniques fall into two categories: lossless and lossy. A lossless compression algorithm guarantees its decompressed output is bit-for-bit identical to the original input. The advantage of perfect reconstruction makes lossless compression useful in certain applications. However, the compression efficiency is not high enough to satisfy all requirements, especially for low target bit rates. The zeroth-order entropy coder, which is also the lossless coder, is a module that attempts to code source symbols at rates approaching the entropy of the source. Huffman coding and arithmetic coding are two basic entropy coding schemes [36]. In practice, for most sources whose probabilities are not exactly the powers of two and whose sequence length and alphabet size are large enough, we achieve rates closer to entropy using arithmetic coding than by using Huffman coding [36], because in Huffman coding, it is infeasible to generate the huge meta-symbol codebook to approach the source entropy. Adapting arithmetic codes to changing input statistics is also much easier than the Huffman case. Thus arithmetic coding is chosen to be the entropy coding method used our modifications to the SFS algorithm.

Lossy compression, on the other hand, means that the output is different from the input. It is widely used when small distortions in the decoded data are tolerable. Intuitively, the lower the required target bit rate, the more compression artifacts appear.

Among the current lossy and lossless compression algorithms, ongoing research indicates that in many applications wavelet-based coding schemes outperform other compression algorithms [50], particularly for compressing medical images [9]. Detailed discussions of wavelets and the wavelet transform are presented in Section 2.1.3.

Measures of performance :

Three basic measurements are used to evaluate the performance of a digital image compression:

- *Compression efficiency*
- *Complexity*
- *Amount of distortion*

Compression efficiency is measured by the compression ratio and is estimated by the ratio of the original image size over the compressed data size. The *complexity* of an image compression algorithm is calculated by the number of data operations required to perform both encoding and decoding processes. Practically, it is sometimes expressed by the number of operations. For a lossy compression scheme, a *distortion measurement* is a criterion for determining how much information has been lost when the reconstructed image is produced from the compressed data. The most often used measurement is the mean square error (MSE), which is often represented by the symbol s^2 or s_d^2 :

$$s^2 = \frac{1}{N}\sum_{n=1}^{N}(x_n - y_n)^2 \qquad 2.2$$

where $\{x_n\}$ are the values of each pixel in the original image, $\{y_n\}$ are the reconstructed image values, and N is the number of pixels. The mean square error is commonly used because of its convenience. A measurement of MSE in decibels on a logarithmic scale is the Peak Signal-to-Noise Ratio (PSNR), which is a popular standard objective measure of the lossy codec. We use the PSNR as the objective measurement for compression algorithms throughout this thesis. It is defined as follows:

$$PSNR(dB) = 10\log_{10} \frac{x_{peak}^2}{s_d^2} \qquad 2.3$$

where x_{peak} is the peak value of the source symbols and s_d^2 is the MSE.

2.1.2 Arithmetic Coding

Arithmetic coding is an entropy coding method for generating variable length codes. It has been increasingly popular over the last decade because of its speed, low storage requirements, and effective compression efficiency. The drawbacks of arithmetic coding include its sensitivity to transmission errors.

In arithmetic coding, a string of symbols is encoded as an interval, which is often called a “tag”, in the range of [0,1). Because the unit interval [0,1) contains an infinite number of intervals, each sequence of symbols can be assigned a unique subinterval. The size of this subinterval is determined by the cumulative distribution function (*cdf*) of the random variable associated with the source [36]. Arithmetic coding can be more clearly understood through an example.

Suppose we want to encode the sequence **a c a a b** with the occurence probability distribution provided in Table 2.1.

Table 2.1 – Symbol probabilities and ranges in the unit interval

Symbol	**Probability**	**Range**
a	0.7	[0,0.7)
b	0.1	[0.7, 0.8)
c	0.2	[0.8, 1.0)

When we encode a sequence, the subinterval that represents the whole sequence is getting narrowed with respect to each symbol's probability. In this example, the first symbol, **a**, falls in the interval of [0,0.7), Therefore, after the first symbol is encoded, the low end of the interval is 0 and the high end is 0.7 for the next symbol. The probability of symbol **c** is 0.2. Instead of the range [0.8, 1) with respect to unit interval, the next range will be [0.8x0.7, 1x0.7). This means that the newly encoded number will have to fall somewhere in the 80th to 100th percentage of the currently established range. Applying this logic will further restrict the range to [0.56, 0.7). The process continues for successive symbols, so that the sequence **a c a a b** is represented by the final interval [0.60802,0.61488). This process is described graphically in Figure 2.1, where the size of the intervals are scaled in order to be visible.

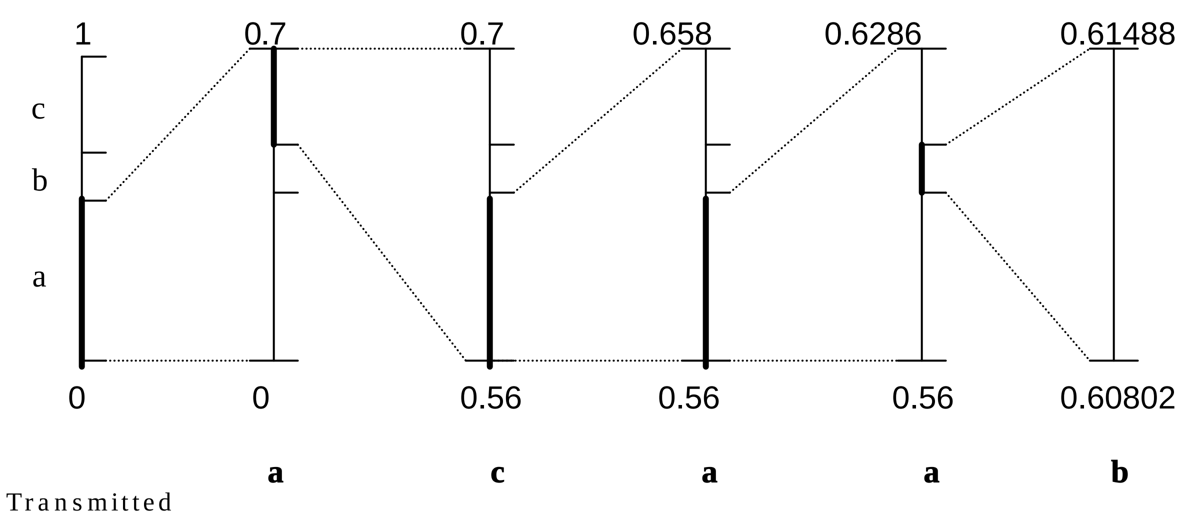


Figure 2.1 – Intervals for the sequence **a c a a b**

As soon as the tag interval lies completely on one side of a “binary fraction line”, a bit of the tag can be sent. The maximum number of bits required to encode the tag is :

$$ceil(\log[\frac{1}{p(x_i)}]) + 1 \qquad 2.4$$

where $p(x_i)$ is the probability of the sequence [36]. Therefore, it is possible to encode the whole sequence of **a c a a b** with

$$ceil(\log[\frac{1}{p(a)\,p(c)\,p(a)\,p(a)\,p(b)}]) + 1 = ceil(7.18758) + 1 = 9bits \qquad 2.5$$

The interval representing the sequence is coded as a string of bits which is defined within the tag. The binary bits are transmitted in order of precision from the most significant bit. In this example, the first symbol **a** lies in the interval of [0, 0.7), which is not confined to either the upper or lower half of the unit interval, so we proceed without transmitting any bit. The second symbol **c** restrains the interval between 0.56 and 0.7, which is included in the upper half of the unit interval, so we send the bit “1”, and the new threshold for the next binary interval, [0.5,1), changes to be 0.75. The third symbol **a** constraints the tag to [0.56,0.658), which falls in the lower half of interval [0.5,1), so we send “0”. The same process continues until the last symbol **b** is encoded. Table 2.2 illustrates the procedure of encoding, showing the transmitted bits (excluding the message termination bits) and the intervals that they are assigned.

To check the answer, the binary fraction .10011 specifies a range between 0.59375 and 0.609375, which is not completely contained within the interval [0.60802, 0.61488). Therefore, we cannot yet finish transmission. Appending binary bits “100” after the output “10011”, we have the output interval of [0.609375, 0.611328125) inside the tag. The output interval will not outstrip the tag once the bit string “10011100” is transmitted. Luckily, in this example, we only need 8 bits to have enough precision instead of 9 bits which is the maximum length to specify the tag of the sequence calculated above.

Table 2.2 – Encoding the sequence **a c a a b**

Symbol	**tag interval**	**Binary interval**	**output**
a	[0,0.7)	[0,1)	-
c	[0.56,0.7)	[0,1)	1
a	[0.56,0.658)	[0.5,1)	0
a	[0.56,0.6286)	[0.5,0.75)	-
b	[0.60802,0.61488)	[0.5,0.75) [0.5,0.625) [0.5625,0.625) [0.59375,0.625)	0 1 1 -

In the decoding procedure, first we meet “1”, which constrains the interval to [0.5,1), this can not tell us exactly what the first symbol is. The second binary bit, “0”, narrows the interval to [0.5,0.75), which still does not define the first symbol of the sequence. After the third bit – “0”, which continues to refine the interval to [0.5,0.625), we can obtain the first symbol, **a**, whose probability range is from 0 to 0.7, completely containing the binary interval. We go on decoding with the same logic and finally get **a c a a b** at the end of decoder. Table 2.3 describes the decoding procedure.

Table 2.3 – Decoding the sequence **a c a a b**

Input	**Binary interval**	**Decoded symbol**
1	[0.5,1)	-
0	[0.5,0.75)	-
0	[0.5,0.625)	**a**
1	[0.5625,0.625)	**c**
1	[0.59375,0.625)	**a**
1	[0.609375,0.625)	**a**
0	[0.609375,0.6171875)	-
0	[0.609375,0.61328125)	**b**

In the above example, we assume that both the encoder and decoder know the length of the message so that the decoder would not continue the decoding process forever. In the real world, we need to include a special terminating symbol so that when the decoder sees this symbol, it stops the decoding process.

In summary, the encoding process is simply one of narrowing the interval of possible numbers with every new symbol. The new range is proportional to the predefined probability attached to that symbol. The output of the encoder are binary bits determined by the sequence tag and the incrementally finer binary intervals with respect to each output. Conversely, decoding is the procedure where the binary interval is narrowed by the input bits, and each symbol is extracted according to its probability and the binary interval.

Arithmetic coding has various advantages over other entropy coding techniques. The most notable one is that it need not generate huge code tables the way Huffman coding does. Another virtue of an arithmetic coder is that it can "adapt on the fly". When the probability of symbols are unknown, both encoder and decoder can estimate the probabilities of the symbols dynamically, based on the changing symbol frequencies in the sequence to be coded. A probability histogram is modeled from part of a sequence that has been seen so far during encoding. There are some techniques to adaptively estimate the probabilities. A general way is as follows[36] :

$$P_A(a) = \frac{b \times x(a) + c}{d \times y + e} \qquad 2.6$$

where $P_A(a)$ is the estimated probability of symbol a in the encoded samples, and y is the number of symbols that have been encoded so far. $x(a)$ is the number of occurrences of a, and A is the input alphabet. Coefficients b, c, d and e are real numbers. Under the condition that the sum of $P_A(a)$ equals to 1 :

$$\sum_{a \in A} P_A(a) = 1 \qquad 2.7$$

Applying (2.7), we have the following equations :

$$d \times y + e = b \times \sum_{a \in A} x(a) + \sum_{a \in A} c = b \times y + n \times c \qquad 2.8$$

where n is the size of alphabet A. For all y and n, Equation (2.7) is satisfied if $b = d$, and then $e = n \times c$. Let $l = \frac{nc}{b}$, we have :

$$P_A(a) = \frac{x(a) + \frac{l}{n}}{y + l} \qquad 2.9$$

where l is thought of as a "trust " parameter, and gives good probability density function for many sources when $l = 1$.

In practice, the exact probabilities of the encoded symbols are usually not available and adaptive arithmetic coding becomes extremely important. The research in this thesis uses adaptive arithmetic coding as the basis for entropy codec.

The basic arithmetic coder uses a single histogram with uniform initialization, and each symbol simply increments the bin count for that symbol by one. However, there are many ways to improve the performance of basic adaptive arithmetic coders. Multiple histograms based on different "context", for instance, could be used to obtain much more skewed distributions. Because the probability density function in each histogram is generated by the symbols belonging to the current "context". This classification reflects the statistical features of the image being coded. Context-based conditional arithmetic coding will achieve shorter code-length, and higher compression ratio, if the context model is selected appropriately to fit the source [60]. There are many ways to form a good context model and they will be discussed in detail in both Section 2.1.4 and Chapter 4.

2.1.3 Subband Coding and Wavelets

In the class of *time domain* coding algorithms, source data is treated as a single full-band signal. In this section, another class of encoding algorithms are discussed, in which the approach is to divide the input signal/image into separate frequency subbands, then quantize and encode each of these subbands separately. This division into frequency components removes redundancy in the input and provides a set of approximately uncorrelated inputs into relatively narrow subbands. One of these *frequency domain* coding schemes is called subband coding [53][56][57]. A wavelet transform is a tree structured subband decomposition with octave band spacing where the lowpass band is the narrowest [22]. Therefore, wavelet transforms are often regarded as a special case of subband coding techniques [50].

Subband coding

Subband coding has been used extensively, first in speech coding [23] and later in image coding [57], because of its inherent advantages namely energy compaction, de-correlation, variable bit allocation among the subbands as well as coding error confinement within the subbands. There are two fundamental components in subband coding: the filter system and the quantization with bit allocation strategy. Figure 2.2 gives a block diagram of a general subband coding system[2].

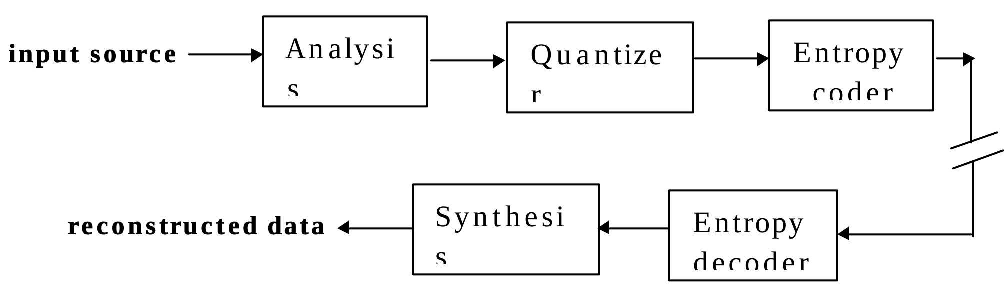


Figure 2.2 – Block diagram of a general subband coding system

The process of separating the signal into frequency components is called filtering. A system that isolates certain frequency components is

---

[2] The analysis filter in the block diagram is critically sampled.

called a filter. The linear phase Quadrature Mirror Filter (QMF), first proposed by Croisier, Esteban, and Galand [2] can result in perfect reconstruction of the input signal without aliasing when channel noise and quantization noise are absent [55].

The two channel filter band [28], illustrated as Figure 2.3, is the core of a subband coding system. It separates the input into frequency bands (filter and downsample), then reassembles these subbands (upsample and filter). In the two-channel one-dimensional system, the inputs are usually decomposed into low frequency and high frequency subbands by the analysis filters L_0 (low-pass filter) and H_0 (high-pass filter), then the filter outputs are decimated by 2. The two "half-length" outputs are filtered by synthesis filters L_1 (lowpass filter) and H_1 (highpass filter) with the upsampling operation by a factor of 2 to reconstruct the original signal. In the center of Figure 2.3, the gap between the analysis system (left part) and the synthesis system (right part) can be filled in by quantization and other signal processing techniques. Proper filter banks should be selected in subband coding to achieve good quality in image coding. Discussion about the constraints on filter designing is given below.

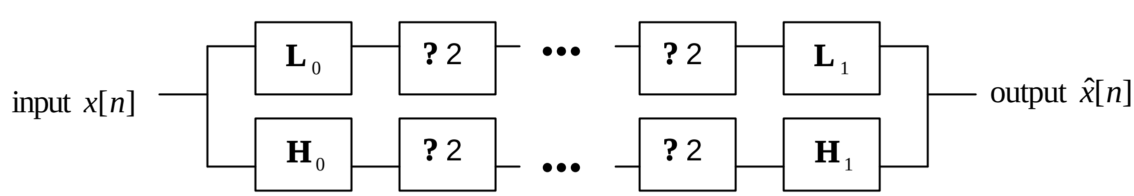


Figure 2.3 – Two channel filter bank

There are many methods for designing Perfect Reconstruction(PR) filter banks. A two channel filter bank gives perfect reconstruction with an *l*-delay when

$$L_1(z)L_0(z) + H_1(z)H_0(z) = 2z^{-l} \quad 2.10$$

$$L_1(z)L_0(-z) + H_1(z)H_0(-z) = 0 \quad 2.11$$

Equation (2.10) is an alias cancellation condition and (2.11) provides perfect reconstruction. Overall, the analysis and synthesis filters are chosen to eliminate the aliasing effects, and to produce a perfect reconstruction in the absence of quantization.

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

$$L_1(z) = L_0(z^{-1}) \quad 2.12$$

and

$$H_1(z) = H_0(z^{-1}) \quad 2.13$$

Expressing Equation 2.10 in terms of L_1 using (2.12) and (2.13) results in the following equation describing the synthesis low pass filter:

$$L_1(z)L_1(z^{-1}) + L_1(-z)L_1(-z^{-1}) = 2 \quad 2.14$$

Equation 2.14 can also be written in terms of the synthesis high pass filter as

$$H_1(z)H_1(z^{-1}) + H_1(-z)H_1(-z^{-1}) = 2 \quad 2.15$$

The relationship between the low pass and high pass synthesis filters is established by applying orthogonality conditions to (2.14) and (2.15)[3].

---

[3] See Vetterli and Kovacevic [52] Appendix 3.A for details.

$$H_1(z) = -z^{-2K+1}L_1(-z^{-1}) \quad 2.16$$

The above equations show that the filter in an orthogonal filter bank are derived from a single prototype filter. Therefore, all the filters have been designed in terms of the low pass synthesis filter L_1, and the lengths of all the filters is even, equaling to $2K$ [53].

A biorthogonal system is less restrictive than an orthogonal system to separate the analysis and synthesis bases. Because 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 PR system. Energy Conservation for a biorthogonal system no longer exists. And the impact for subband coding is that the squared error in the transform domain is no longer exactly equivalent to the squared error in the signal domain.

A PR filter bank described by Figure 2.3, Equation (2.10) and (2.11), implements a biorthogonal series expansion[4]. The analysis basis functions $\tilde{\varphi}_{2k}[l]$ and $\tilde{\varphi}_{2k+1}[l]$ are time reversed even shifts of the analysis filter impulse responses,

$$\tilde{\varphi}_{2k}[l] = l_0[2k-l] \quad 2.17$$

and

$$\tilde{\varphi}_{2k+1}[l] = h_0[2k-l] \quad 2.18$$

The synthesis basis functions $\varphi_{2k}[n]$ and $\varphi_{2k+1}[n]$ are even shifts of the synthesis filter impulse responses,

$$\varphi_{2k}[n] = l_1[n-2k] \quad 2.19$$

and

---

[4] Proof can be found in Vetterli and Kovacevic [52] Section 3.2.1.

$$j_{2k+1}[n] = h_1[n-2k] \qquad 2.20$$

When quantization is required for the subband data, bit allocation, one of the major components of a subband coding system, is applied to efficiently distribute the target bit budget among different subbands. This bit allocation procedure potentially has significant impact on the quality of the final reconstruction.

Researchers have found that the overall distortion of a subband source coding system can be expressed by the sum of separate distortions in each subband, and use a scaling coefficient c_k to compensate for the different sizes and non-energy conservation due to biorthogonal system. Here we use s_k^2 to represent the variance.

$$D = \sum_k c_k s_k^2 \qquad 2.21$$

The optimal bit allocation $\{r_1, r_2, ...r_K\}$ for the entire K subbands minimize the overall distortion :

$$D_{\min} = \sum_{k=1}^{K} c_k s_k^2 \qquad 2.22$$

such that the average coding rate is no larger than the target bit rate R for the whole system

$$\frac{1}{K}\sum_{k=1}^{K} c_k r_k \leq R, \qquad r_k \geq 0 \qquad 2.23$$

There are numerous ways to solve the bit allocation problem: given a target rate, allocate bits to subbands such that the overall distortion will be minimized. Some practical bit allocation schemes can be found in [36], [29], and [21]. In the SFS algorithm, the bit allocation strategy is approached by an optimal representation corresponding to the operating point on the convex hull of the operational distortion-rate function that has a rate close to but does not exceed the target rate [22].

The above one-dimensional subband coding has been very successfully used in speech and audio coding applications [23], and it is easily extended to the two-dimensional case for applications of image compression. The general two dimensional filter banks can be implemented by separable one dimensional filtering, decimation, and interpolation operations on rows and columns respectively. The four resulting subbands after the analysis system are labeled as LL, LH, HL, and HH to denote the split into low and high frequency bands in both the row and column orientations. Figure 2.4 illustrates the conventional description of the four subbands after a single-stage two-dimensional decomposition on image "Barbara". Orthogonal filters (the Daubechies filters of length eight) are used in this example [22]. There is an offset onto each subband as shown in Figure 2.4(b), so that all the symbol values larger than 255 will be chopped to be 255 and all the values less than zero are set to zero. It is noted that most of the energies are concentrated in the LL subband, thus making it look like the original image with only half of the size of each side. In contrast, the high pass subbands seem have much less energy. Furthermore, among the high pass bands, the HL subband has higher energy, which is 82.1% in Barbara. The LH and the HH bands have comparatively less energy than the HL subband, which is 8.6% and 9.3% respectively. In Figure 2.4(b), the energy percentage of the HP bands is also illustrated.

(a) original image (b) subband image

Figure 2.4 – Subbands of “Barbara”

Wavelets and Wavelet Transforms

Wavelets are “small waves” , which have their energy concentrated in time, a property that makes them a natural choice for the analysis of transient, non-stationary, or time-varying signals.

A wavelet basis is generated from a single mother wavelet by simple scaling and translation [14]. In “multiresolution” analysis, the *scaling function* $j(t)$ is related to the lowpass synthesis filter L_1, and the *wavelet* $y(t)$ is related to the highpass synthesis filter H_1. Detail analysis is given in the following descriptions.

The two-dimensional wavelet basis function $y_{j,k}(t)$ is constructed from the function $y(t)$ [5] by the following equation:

---

[5] this function is also called the generating wavelet or mother wavelet.

$$y_{j,k}(t) = 2^{j/2} y\,(2^j t - k) \qquad j,k \in Z \qquad 2.24$$

where Z is the set of all integers. The wavelets, $y_{j,k}(t)$, are a set of real-valued functions.

The two-dimensional scaling function $j_{j,k}(t)$ can also be generated in the same way :

$$j_{j,k}(t) = 2^{j/2} j\,(2^j t - k) \qquad j,k \in Z \qquad 2.25$$

In a wavelet representation, we expand signals in terms of functions that are localized both in time and frequency. Any function $f(t) \in L^2(R)$ can often be represented, analyzed, described, or processed using a combination of the scaling functions $j_{j,k}(t)$ and wavelets $y_{j,k}(t)$ which are orthogonal to each other :

$$f(t) = \sum_{k=-\infty}^{\infty} c_{j_0,k} j_{j_0,k}(t) + \sum_{k=-\infty}^{\infty} \sum_{j=j_0}^{\infty} d_{j,k} y_{j,k}(t) \qquad 2.26$$

The subspaces of L^2 spanned by these functions $f(t)$ in (2.26), g_j , are nested, as described in (2.27), which is also graphically illustrated in Figure 2.5.

$$\cdots \subset g_{-2} \subset g_{-1} \subset g_0 \subset g_1 \subset g_2 \subset \cdots \subset L^2 \qquad 2.27$$

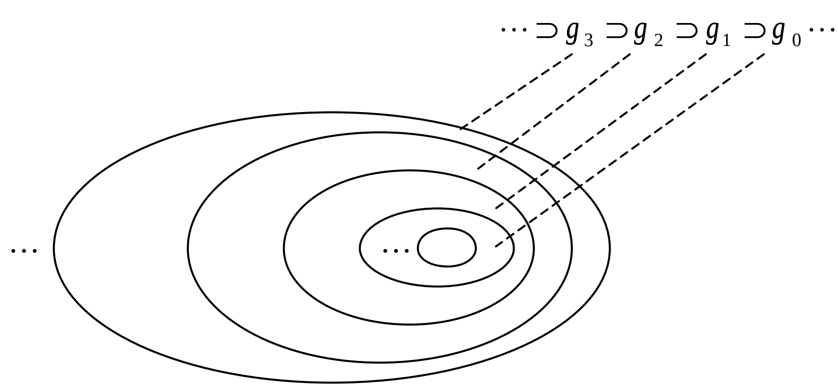


Figure 2.5 – Nested vector spaces spanned by the scaling functions

This nesting means that the space containing high resolution signals will contain those of lower resolution :

$$g_j \subset g_{j+1} \qquad \text{for all} \quad j \in Z \tag{2.28}$$

with

$$g_{-\infty} = \{0\}, \qquad g_{\infty} = L^2 \tag{2.29}$$

The nesting of the spans of the scaling functions indicate that $j(t)$ can be expressed in terms of a weighted sum of shifted $j(2t)$ as

$$j(t) = \sum_n h(n)\sqrt{2}j(2t-n), \qquad n \in Z \tag{2.30}$$

The difference between the spaces spanned by the various scales of the scaling function is described by a set of wavelet functions $y_{j,k}(t)$. The wavelet spanned subspace W_j is defined as following :

$$g_{j+1} = g_j \oplus W_j \tag{2.31}$$

Where $\oplus$ means the direct sum of two subspaces. This means that W_j is the orthogonal compliment of g_j in g_{j+1}. Finally, this gives

$$L^2 = g_{-\infty} \oplus \cdots \oplus W_0 \oplus W_1 \oplus \cdots \tag{2.32}$$

The scale of the initial space is arbitrary and could be chosen at any resolution.

Since the wavelets are located in the next finer space spanned by scaling function, they can be represented by a weighted sum of shifted scaling function $j(2t)$ as

$$y(t) = \sum_n h_1(n)\sqrt{2}j(2t-n), \qquad n \in Z \tag{2.33}$$

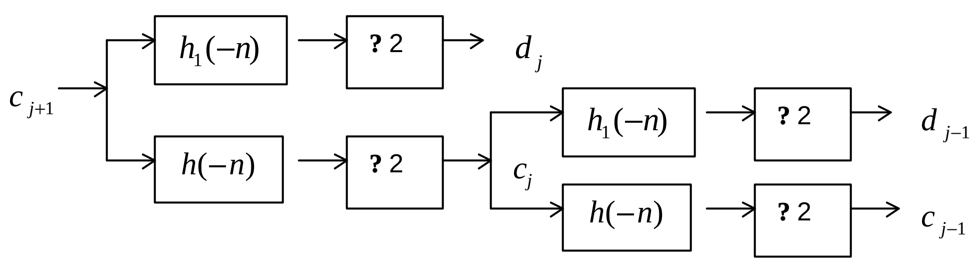
Recalling (2.26), the first summation of scaling function gives a low resolution or coarse approximation of $f(t)$. For each increasing j in the second summation, a higher or finer resolution function is added, which adds increasing detail. The multiresolution concept can be perfectly implemented in this wavelet expansion. Together, the coefficients $c_{j,k}$ and $d_{j,k}$ are called the discrete wavelet transform (DWT) of the signal $f(t)$. They can be calculated by the inner products when the wavelet system is orthogonal.

$$c_j(k) = < f(t), \varphi_{j,k}(t) > = \int f(t)\varphi_{j,k}(t) \quad dt \qquad 2.34$$

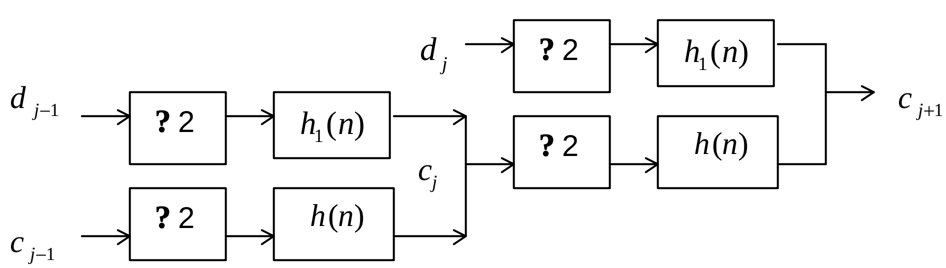
and

$$d_j(k) = < f(t), \psi_{j,k}(t) > = \int f(t)\psi_{j,k}(t) \quad dt \qquad 2.35$$

In compression applications, scaling functions and wavelets are not dealt with directly; instead, the coefficients $h(n)$ and $h_1(n)$ defined in (2.30) and (2.33), and $c_{j,k}$ and $d_{j,k}$ defined in (2.34) and (2.35) need to be considered, and they can be viewed as digital filters and digital signals respectively. The typical two-band filter banks used in the wavelet transform is illustrated in Figure 2.6.



(a) Analysis tree



(b) Synthesis tree

Figure 2.6 – Two-stage, two-band wavelet transforms

Comparing Figure 2.2 of the subband coding and Figure 2.6 of the wavelet transform, we find that they are very similar in filter bank structures except that in the wavelet transform the decomposition has two stages. Actually, the wavelet transform can be viewed as a subset of subband coding, and the filters in wavelet coding are specially designed to tradeoff often conflicting requirements [53][28], such as the short impulse response of both the analysis and synthesis filters, the linear phase of both types of filters, and the orthogonality requirements. All wavelets $y_{j,k}(t)$ should be orthogonal to the scaling functions $j_{j,k}(t)$, and they also should be mutually orthogonal. For an orthogonal filter bank, the analysis bank and the synthesis bank are transposes as well as inverses, but when they are inverse but not necessarily transpose, the filter bank is biorthogonal. From biorthogonal filters, we must expect biorthogonal wavelets. Recalling equation (2.31), W_j and g_j span perpendicular spaces and we have an orthogonal multiresolution. But when the basis is not orthogonal, there is no reason to insist that W_j must be orthogonal to g_j. Therefore, we have a biorthogonal multiresolution and biorthogonal wavelet transform (Detailed description can be found in [28]). Biorthogonality is preferred because it is less restrictive on analysis and synthesis filters.

Having some basic knowledge of wavelet transform, we present some wavelet-based image coding techniques in the following sections.

- ***Some wavelet-based image coding schemes***

Over the past few years, a variety of novel and sophisticated wavelet-based image coding schemes have been developed, among which, Embedded image coding using Zerotrees of Wavelet coefficients (EZW) [31], Set Partitioning In Hierarchical Trees(SPIHT) [4], Space-Frequency

Quantization for Wavelet Image Coding [10], Best Space-Frequency segmentation [22], Compression with Reversible Embedded Wavelets (CREW) [10] , Embedded Block Coding with Optimized Truncation of the embedded bit-streams(EBCOT) [20], Embedded Predictive Wavelet Image Coder [11], Stack-Run image coding [52] and Image Coding using Wavelet Packets [35] are some popular and effective image compression algorithms. Instead of describing all of the innovative and efficient wavelet-based coding methods, we present some basic image coding techniques such as EZW and SPIHT used in the performance comparison, and Image coding using Wavelet Packets algorithms as examples of wavelet-based image compression schemes in this section. JPEG2000 is also included as a reference.

Zerotree coding

There are several ways that subband coding can decompose a signal into various frequency subbands, such as uniform decomposition, octave-band decomposition and adaptive or wavelet-packet decomposition. The octave-band decomposition is a widely used method where the lower frequency subband is split into narrower bands while the high-pass subbands are left without any further decomposition [50]. Each coefficient in course subbands (parent) has four related coefficients (children) corresponding to its spatial position in the finer scale band. Figure 2.7 gives a graphical description of this relationship.

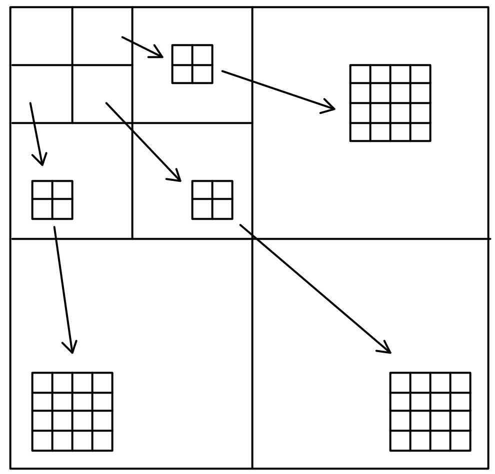


Figure 2.7 – Structure of zerotree

In 1993, Shapiro introduced the concept of "zerotree" trying to find trees of zeros for efficient coding in his elegant EZW algorithm [31]. He applied zerotree coding to successively approximate the wavelet coefficients, providing a compact multi-precision representation of the significant coefficients, having value greater than a given threshold T, and coded them in an embedding manner. The embedded coding technique in his algorithm allows all lower rate codes "embedded" at the beginning of the bit stream, and the bits are "ordered in importance", so that both the encoder and the decoder can stop at any point with the desired rate or distortion. The zerotree is based on the hypothesis that if a wavelet coefficient at coarse scale is insignificant with respect to a given threshold T, then all wavelet coefficients of the same orientation in the same spatial location at finer scales are likely to be insignificant with respect to T. In the EZW scanning order illustrated in Figure 2.8, a zero tree will be defined if the root of the tree and its children are also zero. Many insignificant coefficients at a given threshold T in higher frequency subbands (finer resolutions) can be discarded accordingly. The refinement bits are obtained by comparing with a given threshold T, depending on whether or not they exceed that threshold. The "refined" bits are then transmitted in order of importance, which can be duplicated in the decoder. The entropy coder provides lossless data compression of these bit streams via adaptive arithmetic coding. The entire scheme implements a fully embedded codec, which means that the encoder can terminate the encoding at any point to meet an exact target rate or target distortion, also the decoder can cease decoding at any point in the bit stream and still produce exactly the same image that would be encoded at the same bit rate corresponding to the truncated bit stream.

This algorithm does not need any pre-stored tables or codebooks, training set, or prior knowledge of the image source. Its excellent performance demonstrates the potential for wavelet-based image coding techniques.

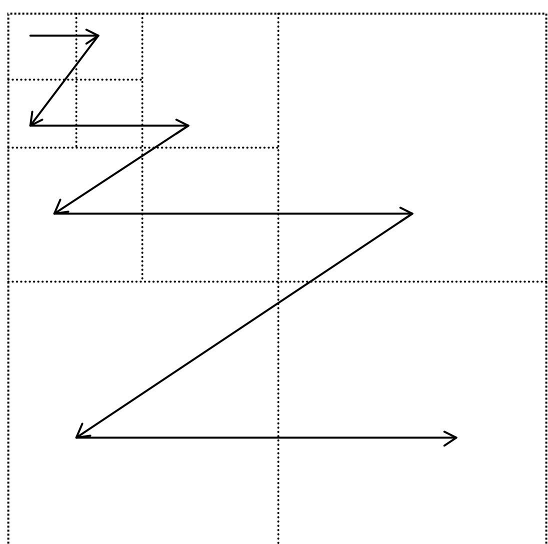


Figure 2.8 – Scanning order of subbands in EZW

Many enhancements have been made on EZW for more robustness and efficiency since its inception. One very popular and superior improvement is the SPIHT algorithm. Said and Pearlman proposed an algorithm based on set partitioning in hierarchical trees to offer an alternative explanation of the principles of EZW operation for better understanding its excellent performance [4]. This novel and more efficient scheme expands the three EZW basic concepts: partial ordering by magnitude of the transformed coefficients with a set partitioning sorting algorithm, ordered bit plane transmission of refinement bits, and exploitation of self-similarity of wavelet transform across different scales of an image. The developments on EZW are the way subsets of coefficients are partitioned and how the significance information is conveyed. The SPIHT compression technique also presents a scheme of progressive transmission of the partial ordering information resulting from comparing coefficient magnitude to a set of octavely decreasing thresholds. In this algorithm, Said and Pearlman adopt a uniform scalar quantizer and claim that the ordering information made this simple quantization method more efficient than expected. The effective progressive transmission scheme on the ordering information achieves good compression performance. The PSNR and execution time of the SPIHT scheme show that this algorithm outperforms those obtained from EZW algorithm [4]. Therefore, this thesis uses SPIHT as one of the basic comparisons to the proposed algorithm.

Both EZW and SPIHT algorithms are applied on an octave subband decomposition obtained by wavelet transforms. The following algorithm tries a different tree structure called the wavelet packet for wavelet decomposition.

Wavelet Packets

Wavelet packets (WP) were first introduced by Coifman, Meyer, Quaker, and Wickerhauser [47][38] as a family of orthonormal (ON) bases for signal compression. The concept of wavelet packets extends the octave-band tree structured filter bank to consist of all possible frequency splits, so that best decomposition topology can be chosen to suit the individuality of different images. Figure 2.9 shows all of the possible representations of a wavelet packet decomposition of maximum depth two, among which the full tree is at the first position from left.

Ramchandran and Vetterli [35] formulate a fast algorithm that prunes the complete tree at a given decomposition depth. The best basis subtree is chosen among the entire library of wavelet packet bases. The problem of bit allocation for the decomposed subbands at a target bit rate is concerned with a set of given admissible quantization choices. The procedure of seeking the best wavelet packet basis and the best quantizer is implemented based on a bit allocation algorithm and a fast rate-distortion based pruning scheme which determines the partition/quantizer combination that minimizes the overall coding distortion subject to a total bit rate budget constraint. The major advantage of the proposed coding scheme is that it adapts the WP tree to the input signal, and its complexity is dramatically reduced by the fast dynamic programming pruning algorithm.

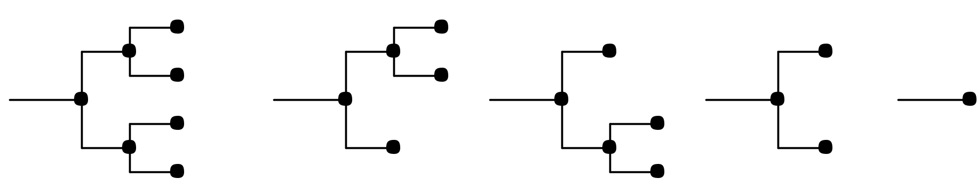


Figure 2.9 – Possible wavelet packet trees, maximum depth=2

JPEG 2000

JPEG2000 is a new image compression standard being developed by the Joint Photographic Experts Group (JEPG). The standard not only aims at providing the best image quality for a given target bit rate, it promises to add various advanced functionalities to support a wide range of applications areas. The new standard addresses the following problems:

- Low bit rate compression (rates below 0.25 bpp)
- Lossless and lossy compression in a single codestream
- Quality and resolution scalability
- Ability to handle large images without tiling (able to handle sizes up to 2^{32}-1)
- Error resilience for transmission in noisy environments
- Region of interest coding
- Ability to efficiently compress computer generated images
- Metadata mechanisms for incorporating additional non-image data as part of the file

In addition, JPEG2000 is able to process data in 256 channels as opposed to only the RGB channels handled by the current JPEG standard. JPEG2000 currently supports two modes: DCT-based coding mode and wavelet-based coding mode. The DCT-based coding mode is primarily provided for backward compatibility with the current JPEG standard. All the new enhancements reside within the 2D discrete wavelet transform based coding mode.

The heart of the wavelet-based JPEG2000 coding mode is the "Embedded Block Coding with Optimized Truncation" (EBCOT) algorithm [20]. The EBCOT algorithms divides each subband into block. Each block is coded independently and a separate bitstream is generated without using any information from other blocks. Each such bitsteam has the property that it can be truncated to a variety of discrete lengths. A post-processing operation is triggered after the entire image has been coded. This operation determines the extend in which each code block can be truncated to achieve a particular target bit rate. The final bitstream is composed of the so called “quality layers”. The basic idea behind this organization is that the substantial overhead required to identify the exact sequence of a large number of block segment is essentially wasted due to the fact that these blocks have very similar rate distortion slopes. Therefore, it makes sense to identify the block truncation points and organize them into layers. The lowest quality layer is first formed. Then, each subsequent layer is formed by optimally truncating the block bitsteams to achieve successively higher target bit rates. The fact that the blocks are coded independently gives rise to the “random access” property of the bitstream.

The EBCOT algorithms uses four types of coding primitives to generate the embedded coding stream for each code block: zero coding, run length coding, sign coding, and magnitude refinement. Run length coding is used in conjunction with the zero coding primitive in order to reduce the average number of binary symbols which must be encoded using the arithmetic coding engine. Sign coding is used at most once for each sample in the block when a previously insignificant symbol is found to be significant during a zero coding or a run length coding operation. Magnitude refinement is used to encode an already significant sample.

It is interesting to note that the JPEG2000 standard is developed with a wide range of applications in mind. The targeted applications areas are the Internet, color facsimile, printing, scanning, digital photography, remote sensing, mobile applications, medical imagery, digital library, e-commerce, and so on. This new standard is currently on the verge of being finalized. With such a large number of new functionalities and potential application areas, JPEG2000 provides a very promising standard for the next generation digital imagery applications.

Summary

The above four wavelet-based coding schemes are all well-known methods for natural image compression. Furthermore, Chiu's research indicates that the zero-tree structure can be improved for ultrasound images and claims that the SFS algorithm is especially suitable for ultrasound images [22].

Arithmetic coding, subband coding and wavelet transforms are the basic coding techniques in image compression. The brief introduction to these coding schemes helps to provide useful insights into the current popular compression algorithms which will be discussed in Chapter 3.

2.1.4 Context-based Entropy Coding

As mentioned in Section 2.1.1, "unconditional" entropy coding is a lossless coding technique in the sense that the output rate approaches the zeroth-order entropy of the input source. Given a finite source $x_1, x_2, ... x_n$, compressing this sequence losslessly requires to process the symbols in some order and try to estimate the conditional probability distribution for the current symbol based on the previously processed symbols [41]. If we use conditional probabilities, we can do better than the zeroth-order entropy. Therefore, the optimal code length of the sequence in bits is calculated as $-\log \prod_{i=0}^{n-1} p(x_{i+1} \mid x_1, \ldots x_i)$. The challenge is to balance the benefit of using extra conditioning information against the performance cost of poor probability estimates.

There are many ways to find good models of conditional distribution for a given data set. One of them is to impose a finite-state structure on the source by assuming the conditional distributions are dependent on the current state of the process. The current state in turn could be defined by a particular neighborhood of nearby pixel values, or in other words the *context*. Conditional probabilities can then be estimated by keeping track of symbol occurrences in each state.

The methods of selecting the order of the model and the specific contexts on which the current symbol is to be conditioned result in two general kinds of context models: causal and non-causal. Figure 2.10 illustrates these two models in a raster-scan images, scanning in row order, from top to bottom, and from left to right in each row.

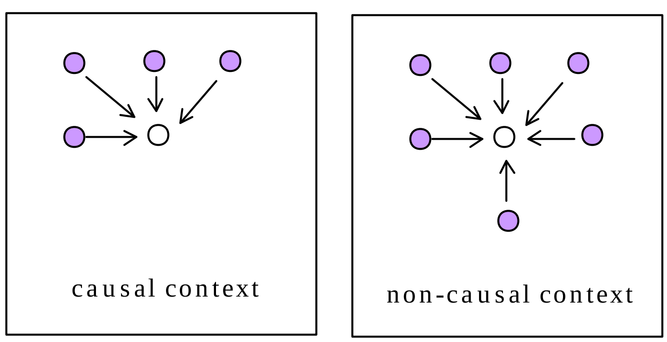


Figure 2.10 – Causal and non-causal context models in raster-scan images

Because the non-causal models include not only causal neighboring pixels but also non-causal ones as its context, it provides more precise context information; The causal model avoids the use of header files, and is a simple scheme in which all the context information relies upon the previous pixels. The less accurate causal model may contain less context information and lead to poor compression results. Ideally, if the context model reflects the conditional distributions of the source data exactly, the conditional entropy coder based on the probability distributions $p(x_{i+1} \mid x_1,...x_i)$ will obtain asymptotically good compression performance. The context size is also an issue for efficient compression. On one hand, the larger the context size the more context information it contains; on the other hand, when the context size is large while the image size is not accordingly large enough, there won't be enough symbols in the image to form an accurate context model, leading to the "data dilation" problem. Therefore, in practice the context size is a tradeoff between compression performance and the image size. Section 4.3 presents detailed discussions of this consideration. In the proposed research, various context models (causal and non-causal) are studied thoroughly along with the optimum number of context states to estimate the optimal conditional entropy coder.

Some sophisticated algorithms for context-based entropy coding are discussed in Section 3.4. They are Wu's Context-based Adaptive Lossless Image Coding (CALIC) [58], Context Selection, Quantization and Modeling (CSQM) [59], Embedded Conditional Entropy Coding Of Wavelet coefficients (ECECOW) [60], and Taubman's multi-resolution prediction model [18].

2.2 Diagnostic Ultrasound Images

Ultrasound refers to sound waves with high frequency (about 18kHz to 10GHz) and ultrasonic imaging is produced by the reflection and refraction of ultrasonic waves at the interface of two different media ([7], [8] and [24]). Background knowledge of ultrasound physics and its image characteristics that follow can be found in Appendix B of this thesis.

When the ultrasound transducer is put against the skin of a patient near the region of interest, it converts the electrical impulse into an ultrasound wave beam which penetrates into the body and reflects back from the surface of the organs inside. Simultaneously, the transducer also acts as a receiver, which translates the ultrasound into electrical signals and detects sound waves as they bounce off from the internal structures and contours of the organs. Different tissues have different reflection waves, causing a signature that can be measured and transformed into an image. These electrical signals are amplified, demodulated, and finally turned into live pictures in an ultrasound system. The general organization of a typical B-scanner is shown schematically in Figure 2.11.

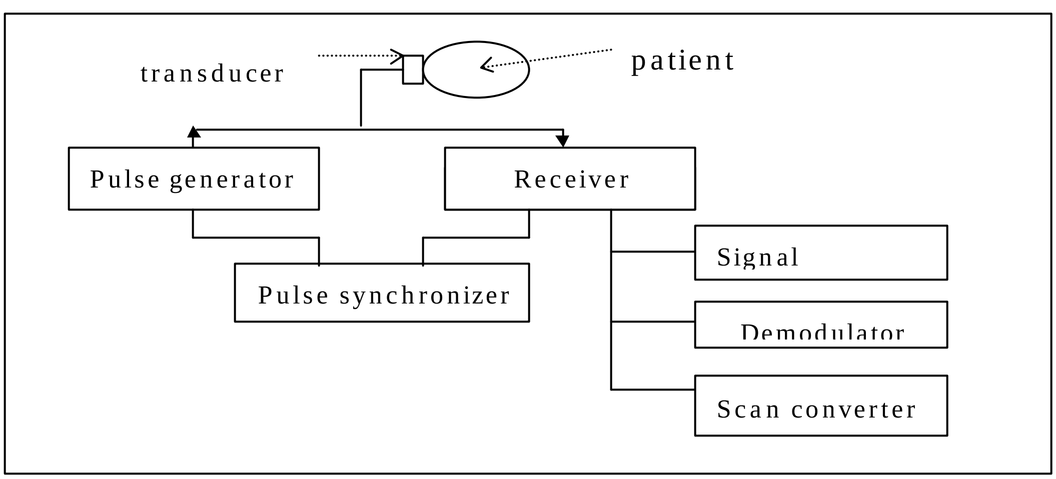


Figure 2.11 – Component of a B-scanner

The electrical signals from returning echo waves can be displayed in three modes: A-Mode (Amplitude mode echo-ranging) , B-Mode (Brightness mode imaging), and M-Mode (Time motion).

A-Mode is the simplest form of ultrasound imaging, which shows only the position of tissue interfaces. In A-Mode, the light spot displayed on the screen moves simultaneously with the regularly produced traveling ultrasound wave. Whenever the sound wave reaches an interface, the transducer receives the echo wave. Accordingly, the light spot gives a short vertical trace of height proportional to the echo strength. Although A-Mode illustrates the basic principles of ultrasound imaging, it is limited in its applications and has been largely replaced by B-Mode imaging or other efficient imaging techniques.

B-Mode imaging electrically displays the amplitude of an echo as the brightness of a light spot on the monitor screen. The scan lines shown on the screen in B-Mode display, corresponding to the pulses, travel in the same direction as ultrasound waves which are a two-dimensional section in either a linear, rectangular or a sector pattern. Small bright dots on the screen will be produced whenever the sound wave encounters a nearly perpendicular boundary; reflected echo waves are stored in a storage display as the data points are acquired. Therefore, an ultrasound image of a tissue surface is made up by those bright dots on each scanning line traveled by each ultrasound pulse. Modern B-scanners enable an automatic scanning in successive ‘frames’, so that the speed is sufficiently fast to demonstrate the motion of tissues.

Figure 2.12 is a general B-Mode ultrasound image from the Imaginis web site[6], it shows the interface of fetal bladder, stomach, heart and the liver/lung.

Figure 2.12 – A typical B-mode ultrasound image

The most prominent artifact in the ultrasound image is known as the "speckle", which is produced by the scattered waves from many small structures or non-smooth surface of the object. When ultrasound waves scatter from the tiny structures or small bumps of a surface, they interfere with each other and generate random echo patterns which are unrelated to the actual pattern of scatters within the organ. The speckles are sometimes diagnostically important, and therefore should be preserved in the images

[6] Imaginis.com is a web site of the Breast Health Specialists for news and information on breast cancer prevention, screening, diagnosis and treatment and related women's health topics. The URL is : http://www.imaginis.net/

[22]. These bright light spots, or “speckles”, are located randomly over the entire image and therefore increase the energy in the high frequency subbands after a subband decomposition. Other special characteristics of ultrasound images are their wedge-shaped scanning areas and the texture information shown in the corner of the images. Those image features make ultrasound image compression different from the coding techniques of other kind of images.

Generally, ultrasound imagery is an important modality in medical imaging techniques that is relatively simple, easy to use, portable, and cost-effective compared to other medical imaging techniques. Because the ultrasound transducer poses negligible radiation risk to both the patient and the examiner, it is believed to be a safe technology. However, ultrasound imaging also has some disadvantages, such as poor quality in terms of details and contrast resolution, the presence of multiplicative high frequency noise, blurring of spatial information perpendicular to the direction of sonic propagation, distortion in regions adjacent to the transducer, and most importantly, the speckle noise that makes subtle differences in the gray levels of different tissues imperceptible and reduces the apparent resolution of the image.

Overall, the characteristics of ultrasound images and the basic image coding techniques introduced in this chapter are the foundation for the research of our proposed algorithm of ultrasound image lossy compression.

Chapter 3

Literature Review

Image compression can be accomplished via many different approaches. A recent study by Erickson et.al [9] compares some modern compression techniques on various medical imaging modalities and concludes that wavelet based coding algorithms, such as SPIHT, outperform other compression schemes including JPEG, which will probably produce blocking artifacts due to the block based DCT transform. In their research, ultrasound images have been pointed out as being difficult to compress partially due to the speckle texture, especially at low bit rates.

Several researchers are examining various avenues to improve efficient ultrasound image coding algorithms, including Wang's progressive image compression [32], Boliek's CREW [10], and Herley's wavelet-based space-frequency segmentation [13][22]. We analyze these algorithms in the next several sections, and then chose our subband decomposition method based upon them. In our research, we concentrate on entropy coding of subband coefficients.

3.1 Embedded Coding Schemes

Progressive transmission or embedded coding is a technique that represents the source by a sequence of binary decisions [31]. The bits are “ordered in importance”, since the embedded code contains all lower rate codes “ embedded” at the beginning of the bit stream. Therefore, both the encoder or decoder can terminate the encoding or decoding at any point, allowing a target rate or distortion metric to be met exactly. Layered systems are often preferred because their progressive property allows users to gradually recover images from low to high quality and stop at any desired bit rate, including near lossless recovery. The successive approximation of images solves the conflict of different requirements on image qualities. The embedded coding schemes have been described in a number of papers. Examples include EZW [31], SPIHT [4], LZC [18], ECECOW [60], CREW [10] , EBCOT [54] and Wang’s progressive compression scheme [32]. EZW, SPIHT and EBCOT algorithms, have been discussed in section 2.1.3, LZC and ECECOW are introduced in Section 3.5. In this section, Wang’s progressive compression technique and Bolie’s CREW are described .

Based upon the basic embedded coding techniques, Wang developed an N-level wavelet-based progressive compression and transmission algorithm combined with a Textual Information Detection and Elimination (TIDE) module that detects textual information in the medical images and tags them for blurring or removal [32]. Progressive coding is used on the wavelet transformation coefficients after the TIDE module. The entropy coder on these subband coefficients of each level is the basic arithmetic coder. The Daubechies-4 wavelet is applied to detect the high frequency variation in the diagonal direction that is indicative of text. Because ultrasound usually has texts of the patient or the examination information at the corner of the image, the TIDE technique is especially useful for detecting and eliminating ultrasound image's textual information from the non-textual area.

However, there are some limitations of this algorithm. First, the text detection algorithm in the TIDE system works on the assumption that the text fonts are smaller in size than major medical objects appearing in the images. This case is not always true in practice. Another important weakness of that scheme is that its goal is to effectively distinguish the areas with and without textual information and try to keep the algorithm simple to achieve a fast computation speed. Therefore, the reconstructed image quality after the texture separation is sacrificed as a tradeoff to obtain a fast algorithm. Because the aim in this thesis is to improve the recovered image quality as much as we can at a given bit rate, the progressive compression method proposed by Wang is not suitable for this application.

Boliek, et.al. [10] developed a compression algorithm with reversible embedded wavelets (CREW) based on Shapiro's zerotree and combined progressive transmission with "Horizon Coding", a technique that uses context based coding to exploit the spatial and spectral information available in the wavelet domain. In Horizon Coding, there are three basic contexts for the embedded encoding of signed integers. The first context is used to encode the event of "zero bit" when the sign bit has not yet been encoded. The second context is used for the "sign bit" if the previous event was non-zero. Context three is used in the case of "zero bit" if the "sign bit" is already encoded. The wavelet transform is localized in both space and frequency. Hence, it is possible for Horizon Coding to use regional contexts in the same band as well as between bands context. The between band spectral contexts can use modeled using the same tree structure as used in Zerotree algorithms. CREW presents superior lossless compression performance than JPEG on deep pixel images (10 bits to 16 bits per pixel), especially for medical imagery (greater than 8 bits deep) and provides lossy compression without many of the artifacts of block-based transform compressors.

The unique feature of progressive transmission algorithms is that it can be used for both lossless and lossy applications due to its embedded quantization.

3.2 Adaptive Quantization and Classification

Chrysafis and Ortega [16] presented a novel approach that combined context classification and adaptive quantization together in the coding of the image subbands. They used the wavelet and apply a uniform threshold quantizer on the subband coefficients. For each symbol, a prediction is made using the nearest three “causal” symbols and one parent-band symbol. An Entropy Constrained Scalar Quantizer (ECSQ)[1] is used on the predictor to classify the current pixel. The number of output points of the quantizer, the context size, is 11 in their experiment [16]. This backward adaptive classification technique, which decides each pixel’s context state determines the probability model for the arithmetic coder. The adaptive quantization in ECSQ is implemented with respect to the rate distortion criterion, $J = D + mR$, where J is the rate distortion cost, D and R represent the distortion and the rate needed respectively, and m

---

[1] This quantizer is described in detail by Sullivan [25].

is a Lagrange multiplier, depending only on the statistics of the image. In this algorithm, the context information includes the quantized coefficients not only in the same subband, but also in its parent subband as well. This context model tries to give a more precise predictor for the subband coefficient. Testing on Barbara, this adaptive quantization method outperforms both EZW and SPIHT at various target bit rates between 0.2bpp and 1.0bpp.

Youngjun Yoo, Ortega and Bin Yu [62] give a different consideration for the quantizer sets in their work. They couple the context classification and the quantization techniques together in their algorithm. First, they separate the subband coefficients into different classes; then they apply different quantization to each class using a bit allocation strategy. The activity of the current coefficient is predicted by a weighted average magnitude of six previously transmitted quantized coefficients. The current pixel is classified by thresholds on the estimated predictor. Unlike the work by Chrysafis [16], the classification thresholds are designed in an iterative procedure aiming at maximizing the coding gain due to classification. The iterative merging algorithm, which tries to merge the pair of classes with smallest gain, converges to a local optimum at each iteration that can uniquely maximize the classification gain. A special class called “zero context” was adopted to separate this kind of context (consisting of all zero-quantized coefficients which contain little information for estimation) from the others. The classification can be formed on a coefficient-by-coefficient basis that overcomes the shortcomings of block-based classification. This classification can split subband coefficients into classes of different variances and shows the advantage of achieving classified regions with arbitrary shapes. After the classification, the quantizer is applied to each classified subband coefficient. Under the assumption of a Laplacian distribution mode, an adaptive Uniform Threshold Quantizer (UTQ) can be derived from the online estimation of model parameters within each class. This algorithm has been proved to be better than SPIHT by experiments at various compression rates between 0.2bpp and 1.0bpp on both Lena and Goldhill. The average improvement in PSNR is 0.24dB.

The adaptive topology is demonstrated to be promising and efficient in lossy image coding. The Space-Frequency Segmentation algorithm [13] is also an adaptive scheme, showing encouraging results on ultrasound images [22]. We now discuss this method.

3.3 Space-Frequency Segmentations

Herley [13] approached a space frequency segmentation technique using balanced wavelet packet trees to represent an image as a set of quantized sub-images. The actual decomposition-quantizer combination was then obtained on the basis of rate-distortion constraints at a given target rate by searching through all possible partitions. The SFS method [13] comes from the single-tree and double-tree wavelet packet algorithms [12]. A single-tree for a segment is pruned using the wavelet packet tree pruning algorithm [35]. For each wavelet transformed subband, quantization is done by a quantizer chosen among a finite set of uniform quantizers and the quantized coefficients encoded by an entropy coder. The winning cost is stored in a binary cost tree. Pruning this cost tree produces the best single-tree basis for the current cost function, and the Lagrangian cost function can be written in “flattened” form as

$$J(l) = D(x) + lR(x) \quad 3.1$$

where l is a Lagrange multiplier, x represents the distortion-rate operating point associated with a particular choice of subtree and quantizer set. $D(x)$ and $R(x)$ are the distortion and rate at x respectively.

The double-tree library allows different wavelet packet single-trees over all binary spatial segmentations of the image. Space-frequency segmentation extents the double-tree segmentation scheme: the binary partition into two one-dimensional segments is replaced by partition into four two-dimensional structures. Each 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 SFS algorithm should generate a better optimal basis than both the single-tree and the double-tree. Its basis must be at least as good as the best double-tree/single-tree basis, because the set of possible double-tree/single-tree bases is a subset of the possible SFS bases. Figure 3.1 compares these three tree structures when the decomposition depth is set to two, where F indicates frequency partition , and S means time partition in one-dimensional signals.

An entropy coder is applied on the quantized subbands to measure the rates of the cost function after the image is segmented. Therefore, the performance of the entropy coder directly influence the winning cost while pruning trees. In Chiu’s implementation of SFS algorithm [22], the basic arithmetic coder is applied on the LL subband, and the stack-run coders are used for high pass subbands. However, the basic arithmetic coder only considers the zeroth-order entropy. The result of this approximation is open for improvement in producing an efficient space-frequency representation with higher-order entropy coder. Chapter 4 will discuss several context-based entropy coders that try to exploit higher-order entropy, and Chapter 5 has the experimental results of these coders.

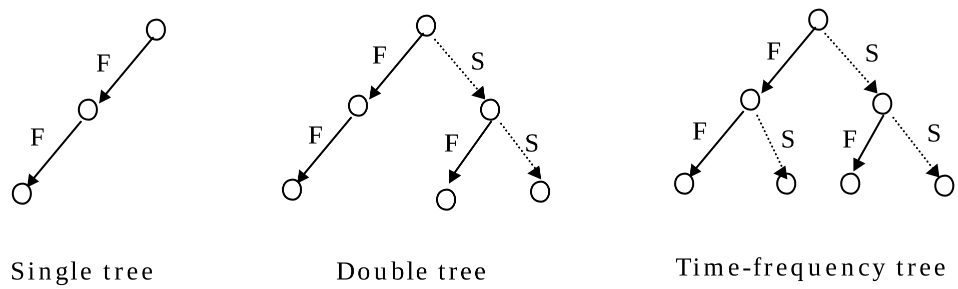


Figure 3.1 –Trees at depth two

An example SFS partition is shown in Figure 3.2. It has maximum decomposition depth of five with a target bit rate of 0.5bpp. The white lines that divide a sub-image means space splitting and the black lines represent the frequency segmentation. If there is no line inside the sub-image, it won't be split by either space or frequency.

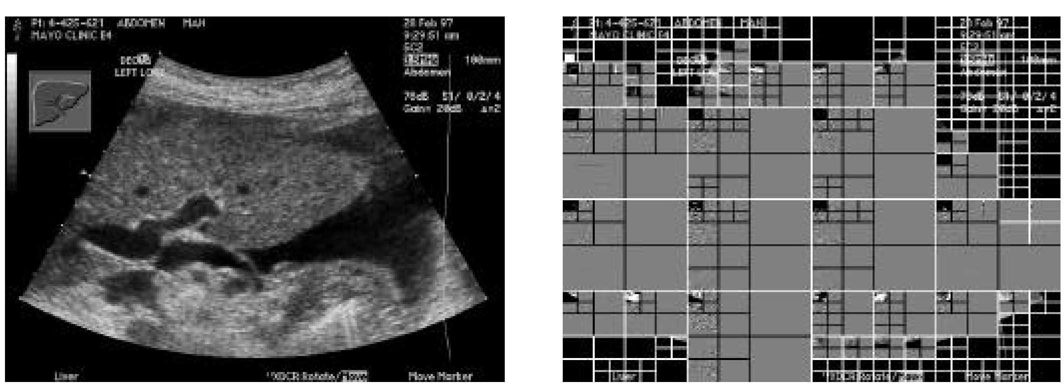


Figure 3.2 – Image U1 and its partition

The PSNR results for ultrasound images using SFS method are higher than those from SPIHT algorithm. Subjective tests using radiologists also confirmed that the SFS images are ranked above the SPIHT ones. Results in Chiu's experiments [22] show that SFS does a better job of preserving both fine detail and low contrast detail. The space-frequency representation indicates that compressing medical ultrasound images needs to treat the background and ultrasound scanned areas differently. SFS does a good job at compressing ultrasound images due to its adaptive nature and its ability to treat different regions differently.

Finally, we choose the SFS technique as the basis of our proposed algorithm according to its better compression performance on ultrasound images compared with the other algorithms, such as SPIHT. From the above discussion, we learn that if the space-frequency segmentation is combined with higher-order entropy coders, such as context-based entropy coders, to exploit dependencies within each subband, more efficient SFS partitions will then be obtained to achieve higher coding gain. We discuss several context-based compression methods in the following section.

3.4 Context-Based Compression Methods

Conditional entropy coding, as we discussed in section 2.1.4, can be effectively combined with lossy image coding techniques to form a more efficient compression system. Various context-based entropy coding strategies have been explored in the recent literature [58, 59, 60, 61].

***Context-based Adaptive Lossless Image Coding* (*CALIC*)**

Context-based adaptive lossless image coding, proposed by Wu, is a well-known lossless image coding scheme [58]. It has four major integrated components: gradient-adjusted prediction (GAP), context selection and quantization, context modeling of prediction errors, and entropy coding of prediction errors. The GAP is a simple, adaptive, nonlinear predictor that can adapt itself to the intensity gradients near the predicted pixel. For each pixel $x[i, j]$, Wu uses *n, w, ne, nw, nn, ww, nne* to denote north, west, northeast, northwest, north-north, west-west and north-northeast pixels respectively that form a context model as shown in Figure 3.3.

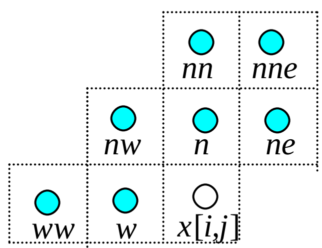


Figure 3.3 – The GAP predictor

In the prediction algorithm, the predictor becomes either pixel *n* or *w* when there is a reasonable likelihood that a vertical or horizontal edge exists; otherwise, it is selected from several choices according to the local smoothness, which needs to be estimated. When the samples of *n, w, ne, nw, nn, ww* and *nne* are known to both the encoder and decoder, the parameters d_h and d_v can be estimated by the following equations:

$$d_h = |w - ww| + |n - nw| + |ne - n| \quad 3.2$$

$$d_v = |w - nw| + |n - nn| + |ne - nne| \quad 3.3$$

In the algorithm, d_h and d_v are introduced to represent the horizontal and vertical gradients for each image sample during encoding. Detected by several thresholds for d_h and d_v, which will be described in detail in Section4.1.3, the magnitude and orientation of the edges in the horizontal and vertical directions can be approximated and then predictors can be adjusted on the fly. Wu uses GAP for his context modeling to predict the continuous-tone values for the pixels to be coded. An energy estimator of the predictor error is then calculated and quantized to classify the errors according to their variances. A binary parameter called the spatial texture pattern is also formed by comparing the local neighborhood of pixel values with the predictor. Both the quantized prediction error energy and the spatial texture pattern form the compound modeling contexts. Finally, a conditioned adaptive arithmetic coder is run on the prediction errors. Thus entropy coding of errors using estimated conditional probability improves coding efficiency. Comparing with the lossless JPEG algorithm[2], the average compression ratio for CALIC is 12% better on a set of ISO test images.

Context Selection, Quantization and Modeling (CSQM)

Wu extended CALIC to the lossless compression of continuous-tone images [59]. Instead of using his causal context model, the GAP, Wu tried a non-causal context model in this algorithm, which completely forms a spatially enclosed 360° model surrounding the modeled pixel. It optimizes the prediction in a more precise way for a raster scan ordered image compression technique than a causal context model does. The adjacent and enclosing modeling is done in three passes. Each pass uses an

---

[2] It is the old version of lossless compression in JPEG.

interlaced sampling of the original image. Similarly to CALIC, a compound context in CSQM is determined by both the quantized prediction error energy and the spatial texture pattern. Detailed estimation will be given in the following description of this section. Conditional adaptive arithmetic coding is then used to encode the prediction error. The three passes of interlaced sampling scheme and the resulting prediction and modeling contexts are shown in Figure 3.4, where the shaded circles represent contexts and the hollow ones stand for the currently coded samples.

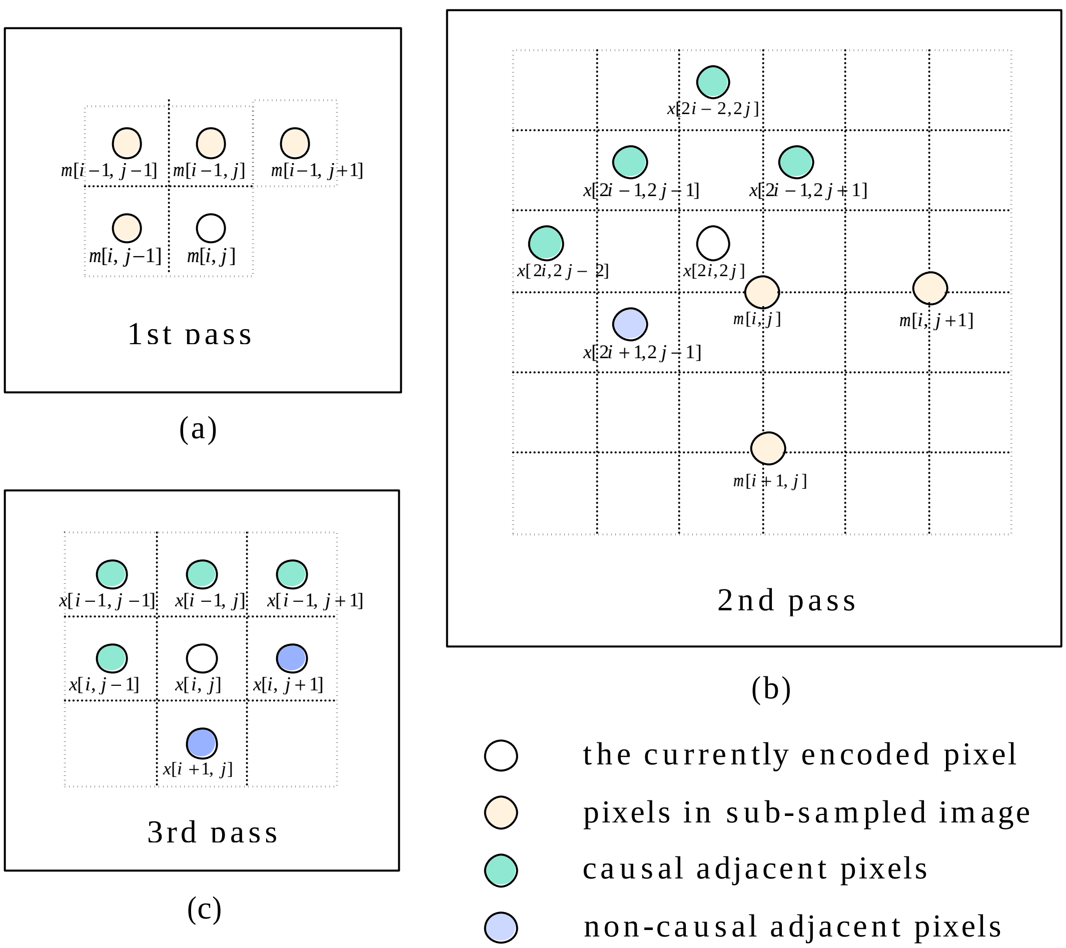


Figure 3.4 – Context modeling of the three passes

Each of the three interlaced predictive coding schemes uses an interlaced sampling of the original image. The predictor coefficients for each pass are determined by linear regression with a training set [59].

Before the first pass, a sub-sample is constructed for future prediction. The sub-sampled W/2 × H/2 image denoted by $m[i,j]$, is obtained from the following equation:

$$m[i,j] = \left\lfloor \frac{x[2i,2j] + x[2i+1,2j+1]}{2} \right\rfloor \qquad 3.4$$

where $m[i,j]$ is the diagonal mean of the two diagonally adjacent pixels that are encoded in the first pass.

In the following three equations describing the three passes, $\hat{x}$ presents the predictor. The first pass prediction is as follows:

$$\hat{x} = \frac{m[i,j-1] + m[i-1,j]}{2} + \frac{m[i+1,j-1] - m[i-1,j-1]}{4} \qquad 3.5$$

The second pass uses sub-sampled image $m[i,j]$ and some causal adjacent samples as the prediction contexts to predict $WH/2$ pixels $x[2i,2j]$:

$$\begin{aligned} \hat{x} = {} & 0.9m[i,j] + \frac{x[2i-1,2j+1] + x[2i-1,2j-1] + x[2i+1,2j-1]}{6} \\ & -0.05(x[2i-2,2j] + x[2i,2j-2]) - 0.15(m[i+1,j] + m[i,j+1]) \end{aligned} \qquad 3.6$$

From Equation (3.6), we can see that $x[2i,2j]$ can be predicted at higher accuracy than predicted from a single-pass causal context model. This is because the predictor has some non-causal adjacent samples, the sub-sampled image, as its context. Once $x[2i,2j]$ is reconstructed, the decoder can set $x[2i+1,2j+1]$ to be $2m[i,j] - x[2i,2j]$ without receiving any extra information.

On the basis of the first two passes, the third pass encodes the remaining half of the original W×H image $x[2i+1,2j]$ and $x[2i,2j+1]$. A larger 360^0 type contexts is used and linear predictor is as follows:

$$\hat{x} = \frac{3}{8} x[i-1, j] + x[i, j-1] + x[i+1, j] + x[i, j+1] - \frac{x[i-1, j-1] + x[i+1, j-1]}{4} \qquad 3.7$$

The main purpose of the three-pass interlaced sampling scheme is to form the adjacent 360^0 type modeling contexts for as many pixels as possible.

After the predictor is determined, the context pattern and the error discriminant for each pixel is calculated for context modeling. Around the current coding pixel $x[i, j]$, a binary number $t = t_K \ldots t_1$ of K bits, obtained from its neighbors $x_1, x_2, \ldots, x_K$, is named as *spatial texture pattern* and specified by the following equation:

$$t_K = \begin{cases} 0 & if \quad x_k \geq \hat{x} \\ 1 & if \quad x_k < \hat{x} \end{cases} \quad 1 \leq k \leq K \qquad 3.8$$

Besides the spatial texture patterns t, the error strength discriminant Δ also forms part of the context for each encoded sample $\hat{x}$. This parameter is the least-squares estimator of $|e|$, where $e = x - \hat{x}$, defined as:

$$\Delta = \sum_{k=1}^{K} w_k \, | x_k - \hat{x} | \qquad 3.9$$

where w_k is the coefficient chosen by linear regression to minimize the following error measure over the training set **S**:

$$\sum_S (|e| - \Delta)^2 = \sum_S (| x - \hat{x} | - \sum_{k=0}^{K} w_k \, | x_k - \hat{x} |)^2 \qquad 3.10$$

Once Δ is estimated over a training set, it is quantized by a quantizer that minimizes the conditional entropy of the prediction error. In an off-line design process, standard dynamic programming is used as: $0 = q_0 < q_1 < \cdots < q_{L-1} < q_L = \infty$, where q_l is the quantizer output point. In practice, $L = 8$ is found to be appropriate. The context is then formed by a combination of the quantized error strength discriminant Δ and the spatial texture patterns *t*. Testing on a set of 8-bit natural JPEG images, this algorithm achieves an average 13.4% better compression rate than the standard JPEG lossless compression scheme.

Embedded Conditional Entropy Coding Of Wavelet coefficients

Wu and Chen [60] have also studied the problem of context modeling for the binary symbol streams generated by the EZW image coder discussed in Section 2.1.3. This combination of adaptive context modeling and conditional entropy coding of wavelet coefficients in the wavelet frequency-time domain is reported to be the best performing compression scheme in the literature as of 1997. Working on Lena and Barbara, between 0.25bpp and 1.0bpp target bit rates, they have achieved on average a 5.33% increase on PSNR value than EZW and 2.62% than the SPIHT algorithm. They suggested a new technique named Embedded Conditional Entropy Coding Of Wavelet coefficients (ECECOW), which encodes a uniformly-quantized wavelet coefficients bit plane by bit plane, scanning from the most to the least significant bits to obtain an embedded binary code stream. The contexts are formed from the current EZW coefficient contexts modeled by the spatial domain adjacencies and the related symbols of the previous dominant pass, together with parent-band relevant pixels.

Multi-Layer Coding

Another example for combining the progressive coding technique and context modeling schemes can be found in David Taubman and Avideh Zakhor's multi-rate subband coding [18]. They have designed multi-layered quantization and Layered Zero Coding (LZC) technique, exploiting the spatial correlation among and between the subbands and enhancing the coding efficiency for high frequency subbands where zeros are the most frequent symbol. They consider using both zigzag scanning and local and wider scale conditionings based on the samples adjacent to the pixel being coded. For those samples that are on future scanning lines, they use values of the previous quantization layer in the multi-layered quantization structure. To take advantage of 2-D dependencies amongst the elements of zero symbols, they also adopt conditional arithmetic coding algorithm for entropy coding of symbols.

The idea of multi-rate and multi-layered quantization in [18] has been applied in JPEG2000 [20], and will also be chosen as the "base" of our non-causal context model entropy coding in Section 4.2.2. Both the encoder and decoder contain two parts: layered zero coding and layered PCM/DPCM (Pulse Code Modulation/Delta Pulse Code Modulation).

In multi-layered quantization [18], each subband is quantized by different quantizers $Q_1, Q_2, ...Q_N$ in different quantization layers $L_1, L_2, ...L_N$. Then for each layer L_l, the quantized symbol sequence is encoded into the current layer L_l on the condition of the previous symbols in the same layer and the previous quantization layers $L_1, L_2, ...L_{l-1}$. Comparing with ECECOW, which codes the quantized wavelet coefficients bit plane by bit plane, this multi-rate scheme will code symbols for each quantization layer instead of each bit plane in ECECOW. The number of layers chosen by Taubman and Zakhor is nine by experiments in their research. Figure 3.5 depicts this multi-layered quantization structure.

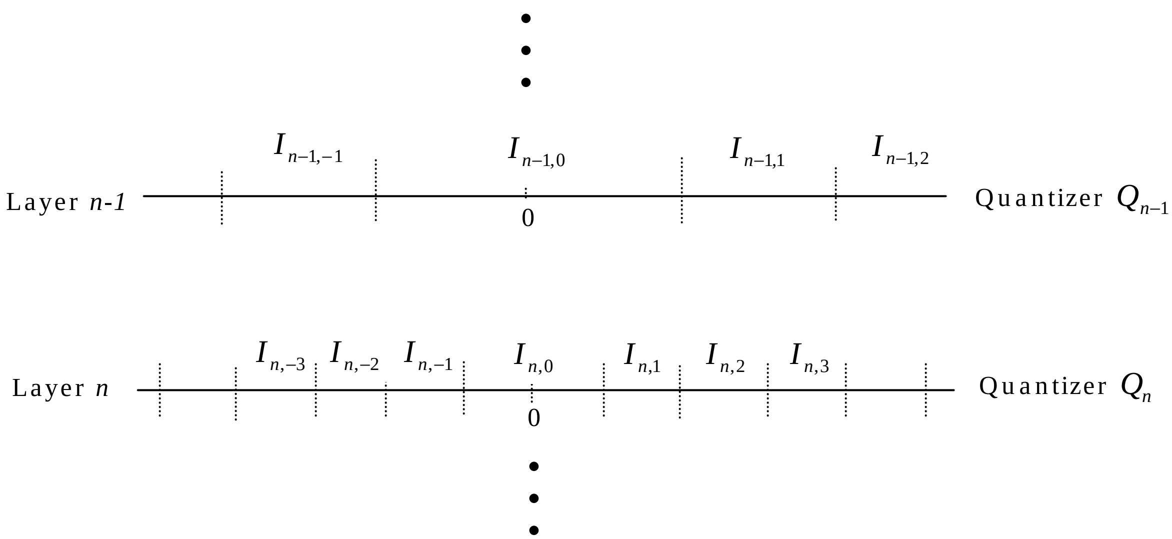


Figure 3.5 – Illustration of layered quantization

Taubman [18] proved that the total number of bits required to encode layers $L_1, L_2,...L_N$ is approximately the same as the number of bits required to encode the output of quantizer Q_N alone. This is their multi-layer coding efficiency goal and can be achieved for layered PCM coding.

In Taubman's algorithm [18], layered PCM is applied to all high frequency subbands and layered DPCM is applied to the low frequency subband. These two techniques are used for encoding the subband coefficients combining with layered zero coding described as below.

Layered zero coding :

The object of zero coding is to effectively exploit the spatial correlation in the subband. A sequence of $s_n[i, j]$ for layer n is created simply via the so-called white and black run lengths coding technique where

$$s_n[i,j] = \begin{cases} 1, & if \quad Q_n(x[i,j]) \neq 0, \quad for \quad PCM \\ & or \quad if \quad dQ_n[i,j] \neq 0, \quad for \quad DPCM \\ 0, & \qquad\qquad\qquad\qquad else \end{cases} \qquad 3.11$$

In order to take advantage of 2-D dependencies among the elements of $s_n[i,j]$, a sequence of "conditioning terms" denoted by $k_n[i,j]$ is introduced. In this algorithm, the basic idea is to first consider the values of the samples adjacent to $x[i,j]$, which are collectively referred to as the local scale conditioning, $k'_n[i,j]$. Only if all of these samples are found to be zero, is wider scale conditioning is resorted to. $k_n[i,j]$ is a combined consideration of local and wide scale conditionings.

Figure 3.6 provides the framework for understanding the local and wide scale approaches respectively. In the figure, the hollow circles represent the coded symbol $x[i,j]$, others stand for the contexts, among which the shaded ones are available at the current quantization layer, and the solid ones are available at previous quantization layer.

The local conditioning information $k'_n[i,j]$ is summarized as a binary label according to:

$$k'_n[i,j] = a + 2b + 4g + 8h + 16V + 32r \qquad 3.12$$

and

$$\begin{aligned} a &= s_n[i, j-1] \\ b &= s_n[i-1, j] \\ g &= s_n[i-1, j-1] \quad i \quad s_n[i-1, j+1] \\ h &= s_{n-1}[i, j+1] \\ V &= s_{n-1}[i+1, j] \\ r &= s_{n-1}[i+1, j-1] \quad i \quad s_{n-1}[i+1, j+1] \end{aligned} \qquad 3.13$$

where i denotes logical "or".

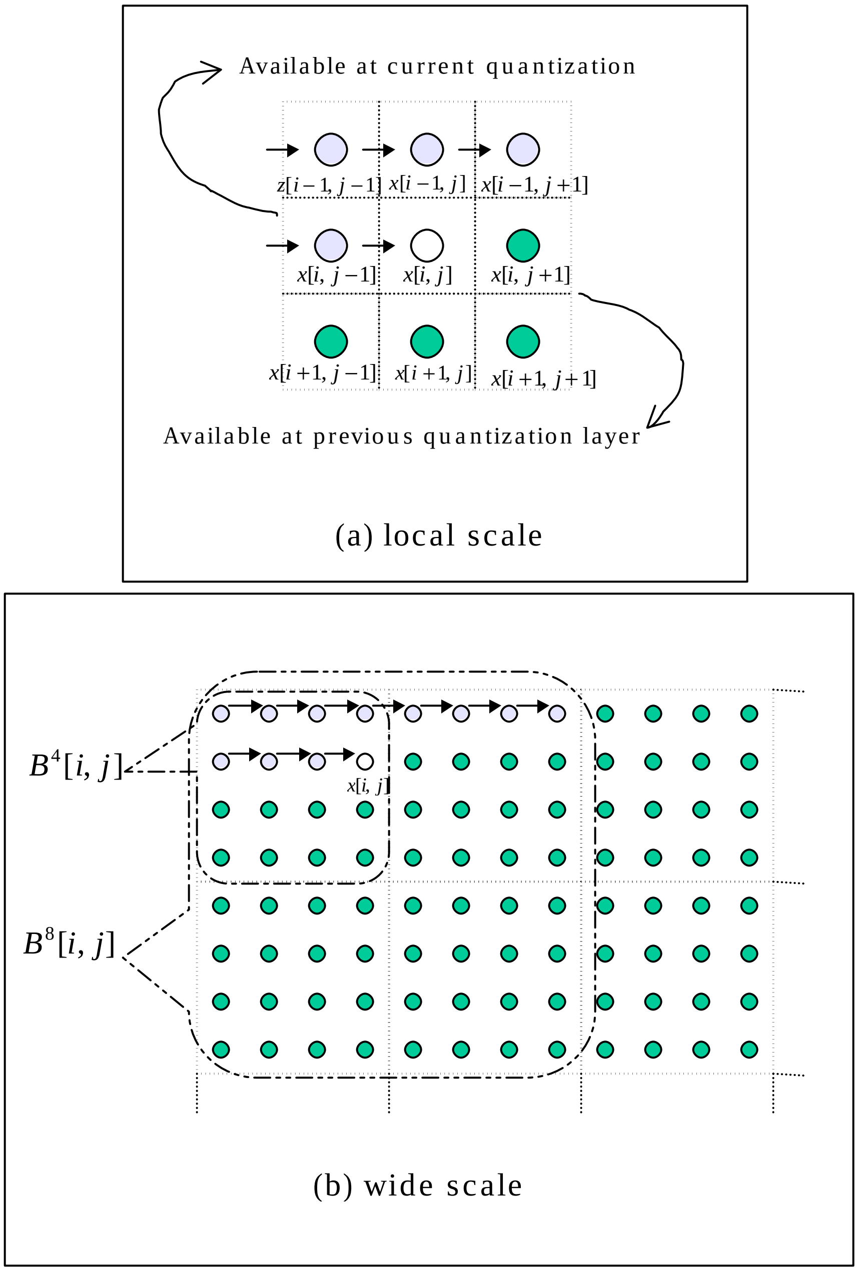


Figure 3.6 - Local and wide scales in Layered Zero Coding

From equation (3.12) and Figure 3.6(a), we can see that the most immediate four neighbors of $x[i, j]$, $x[i, j-1]$, $x[i, j+1]$, $x[i-1, j]$ and

$x[i+1, j]$, corresponding to the first, second, fourth and fifth terms in the right hand side of (3.12) determine the value of the first, second, fourth and fifth bit of the binary representation of $k'_n[i, j]$. The corner neighbors, $x[i-1, j-1]$, $x[i-1, j+1]$, $x[i+1, j-1]$ and $x[i+1, j+1]$, are grouped into pairs, corresponding to the third and sixth terms in the equation, via the logical "or" operator. These pairs determine the remaining two binary digits, the third and sixth bits of $k'_n[i, j]$.

The wide scale conditioning terms, called $B_n^4[i, j]$ and $B_n^8[i, j]$ are defined using the s_n values in blocks of pixels, as shown in Fig 3.6(b):

$$
\begin{aligned}
B_n^4[i, j] &= \max_{k,l \in B^4[i,j];k,l<i,j} s_n[k,l] \; i \; \max_{k,l \in B^4[i];k,l>i,j} s_{n-1}[k,l] \\
B_n^8[i, j] &= \max_{k,l \in B^8[i,j];k,l<i,j} s_n[k,l] \; i \; \max_{k,l \in B^8[i,j];k,l>i,j} s_{n-1}[k,l]
\end{aligned}
\quad 3.14
$$

Finally the local and wide scale terms are combined to form the ultimate sequence of conditioning terms, $k_n[i, j]$, another binary label according to

$$
k_n[i, j] = \begin{cases} 2 + k'_n[i, j] & if k'_n[i, j] \neq 0 \\ \max\{2B_n^4[i, j], B_n^8[i, j]\} & if k'_n[i, j] = 0 \end{cases} \quad 3.15
$$

Then the sequence of $s_n[i, j]$ is encoded on the condition of $k_n[i, j]$ for each sample on each quantization layer. It was found in practice that most of the coding gain is obtained from conditioning coding of $s_n[i, j]$ relative to the local scale term, $k'_n[i, j]$ [18]. The incorporation of wide scale terms, $B_n^4[i, j]$ and $B_n^8[i, j]$, reduces the bit rate by only a few percent, but with negligible impact on the computational demand of their algorithm. Therefore, they form the context using (3.15).

From the successive quantization layer (see Figure 3.5) we can see that once a sample has shown to be nonzero in quantization layer L_{n-1}, there is no need to code this sample, as being non-zero, in quantization layer L_n.

Layered PCM/DPCM:

For each quantization layer, the subband samples whose $s_n[i, j]$ equal to one will be coded by PCM (for high frequency subband) or DPCM (for low frequency subband).

First, the quantizer used in each layer will be examined. For high frequency subbands, the sample values are comparatively small, and zero is the most frequent sample. So the quantizer designed for layered PCM is proposed to have a dead zone, a large quantization interval around $x = 0$, to increase the efficiency of zero coding techniques. For quantizer Q_n, the interval $I_{n,k}$ is defined as follows:

$$I_{n,k} = \begin{cases} (-q_n, q_n) & \text{if} \quad k = 0 \\ [q_n + (k-1)\Delta_n, q_n + k\Delta_n) & \text{if} \quad k > 0 \\ (-q_n + k\Delta_n, -q_n + (k+1)\Delta_n] & \text{if} \quad k < 0 \end{cases} \qquad (PCM) \qquad 3.16$$

Taubman and Zakhor have proved that the multi-layer coding efficiency goal, mentioned above, holds if and only if the following two conditions are satisfied :

$$\Delta_{n-1} = K_n \Delta_n \qquad 3.17$$

$$q_n = q_{n-1} - K_n^{'} \Delta_n \qquad 3.18$$

where K_n and $K_n^{'}$ are arbitrary positive integers. Equation(3.17) indicates that the quantization step size must be divided by an integral factor from layer to layer. In practice, they chose the smallest useful factor, $K_n = 2$. They also halved the dead zone threshold, q_n, in each successive quantization layer for consistency and ease of implementation. One easy way to set this is to set $q_n = \Delta_n$, which implies that $K_n^{'}=1$. Overall, they obtained the following equation, which means that each quantizer has a dead zone twice as large as the other quantization intervals :

$$I_{n,k} = \begin{cases} (-\Delta_n, \Delta_n) & if \quad k=0 \\ [k\Delta_n, (k+1)\Delta_n) & if \quad k>0 \\ ((k-1)\Delta_n, k\Delta_n] & if \quad k<0 \end{cases} \qquad (PCM) \qquad 3.19$$

For the low frequency subband, an average value of that subband, a_n, is associated with the quantizer in each layer:

$$Q_n(x) = \left\langle \frac{x - a_n}{\Delta_n} \right\rangle \qquad 3.20$$

where $\langle \cdot \rangle$ denotes rounding to the nearest integer and there is no longer dead zone in the quantizer. So the quantization intervals corresponding to the quantization operation are given by the following equation:

$$I_{n,k} = \begin{cases} (a_n - \frac{\Delta_n}{2}, a_n + \frac{\Delta_n}{2}) & if \quad k=0 \\ [a_n + (k-\frac{1}{2})\Delta_n, a_n + (k+\frac{1}{2})\Delta_n) & if \quad k>0 \\ (a_n + (k-\frac{1}{2})\Delta_n, a_n + (k+\frac{1}{2})\Delta_n] & if \quad k<0 \end{cases} \qquad (DPCM) \qquad 3.21$$

After quantization, an integer valued refinement function, l_n is defined by $l_n(x) = Q_n(x) - K_n Q_{n-1}(x)$. It is proved in [18] that $0 \le l_n(x) < K_n, \forall x$. So l_n should be either zero or one when K_n is set to two as discussed above.

In PCM, the quantized values of the subband samples $Q_n[i,j]$ are coded by the arithmetic coder conditioned on the reference refinement value $l_n(x[i,j])$ and the quantized value, $Q_{n-1}[i,j]$, in the previous quantization layer. Because there are two possible values for each l_n, the context number for the n^{th} quantization layer L_n is $2M_n$ where M_n is the maximum output number of quantizer Q_n. However, in the DPCM case, the integer sequence $dQ_n[i,j]$ will be encoded and decoded instead of $Q_n[i,j]$. $dQ_n[i,j]$ is given by $dQ_n[i,j] = Q_n(x[i,j]) - Q_n(\tilde{x}[i,j])$, where the reference sequence $\tilde{x}[i,j]$ for the sequence of current sample values $x[i,j]$ is simply given by $\tilde{x}[i,j] = x[i,j-1]$, with the edge symbols coded independently. In particular, the arithmetic coder for DPCM is conditioned on both $l_n(\tilde{x}[i,j])$, and $dQ_{n-1}[i,j]$, and the context number for DPCM is exactly the same as PCM presented previously.

3.5 Summary

We have discussed some current lossy image coding algorithms in this chapter, especially for ultrasound images. We also described several context-based image compression techniques that try to exploit the higher-order entropy of the images. Among these algorithms, there are several methods to create a context model. One is to quantize the predictor of the current pixel, and the output of the quantization produces the context. Chrysafis and Ortega [16] adopt this method. Another way to form the context is combining and quantizing several prediction error related values. Wu's CALIC [58] and CSQM [59] algorithms are examples for this context modeling technique. Applying multi-layered quantization and using symbols in both the current and previous layers to form a context is the way Taubman and Zakhor [18] used. We will apply the above three context modeling techniques in our context-based entropy coders in Chapter 4.

It is impossible to summarize all of the state-of-the-art context-based entropy coding algorithms within this chapter. The discussion on context models is intended to shed some light, and offer encouraging suggestions on the proposed context-based entropy coding scheme researched in this thesis.

Chapter 4
Experiments on Context-Based Entropy Coding

Context modeling is emerging as a promising approach to compressing continuous-tone images, including ultrasound images. In this chapter, we focus on the study of context modeling the quantized subband coefficients based on space-frequency signal decompositions. Proof of concept experiments are designed for choosing an efficient context model for the proposed algorithm.

There are several ways to model a data source using contexts. Three steps can be basically identified in the search for a good model [41]:

1. Choose a suitable family of model classes to represent the data.
2. Select a model order.
3. Decide on the parameters that fit the data given the model and its order.

As discussed in Section 2.1.4, the coders that implement conditional arithmetic coding first classify the data and then use a set of different probabilities for each class. Accordingly, the more precisely the classes represent the distribution structure, the more efficient the coder will be. There are many ways to form a context. Referring to Chapter 3, in Wu's CALIC [58] and CSQM [59] algorithms, the combination of the spatial pattern, which is a binary sequence for charactering spatial structure, and the quantized prediction error energy level is applied to form conditioning states in entropy coding of the wavelet coefficients. Chrysafis and Ortega [16] applied quantization on the predictors to classify the quantized wavelet coefficients. While the context in Taubman's layered zero coding scheme [18] is determined by the local and wide scale conditioning information (see Section 3.5 for a detailed description). In our proposed research, because the statistical structures of the quantized SFS subband coefficients are quite different from each other, and it is hard to use a single model to describe them. Different subbands need different quantizers with different quantization intervals. It is impractical to transfer these statistical properties of each subband during coding. Therefore, under a conservative consideration that all the distributions of the predicted subband coefficients are the same, a uniform quantizer is then applied on the predictors of wavelet coefficients to form contexts. We experimented with various predictors in order to find the most suitable one. The number of context states, which is also the output of the quantizer, will be discussed later at the end of this chapter.

In general, model selections are made using two general categories of context models: causal and non-causal context models. In causal context modeling algorithms, the predictor is estimated from a combination of the neighboring coefficient values. Quantization of these predictors is then performed to obtain the contexts. For the non-causal context models tested in our experiments, the compound context modeling technique in Wu's CALIC and CSQM is applied. Contexts generated from both current and previous quantization layers are also tested in our multi-resolution algorithm described in Section 4.2.2.

4.1 Causal Prediction-based Context Models

A causal context model uses “past” values to form the context. In a raster scanned image, the causal context model contains the neighbors to the left and at the top of the symbol being coded. No side information is required to decode the sequence of bits, because it has reconstructed the previous symbols to obtain the context of the current decoded symbol. Predictions are generated through the linear combination of these “past” values in a context model. These predictions can then be coarsely quantized to form a small number of contexts. There are various methods to form a causal context model. Several models are introduced as follows.

4.1.1 The Simplest Causal Model (SP1)

In this very simple context modeling technique, since only the left adjacent sample $x[i, j-1]$ is considered as the predictor for the currently coded sample, $x[i, j]$. Figure 4.1 gives an illustration of this model.

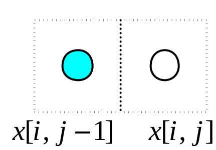


Figure 4.1 – The simplest causal context model

This kind of prediction model is only concerned with the horizontal relationship between samples, whereas in most subbands, both the horizontal and the vertical dependencies exist. Therefore, both of these are taken into account in the next section to improve the context model.

4.1.2 Modified Simple Causal Model (SP2)

The simple causal model applied here takes into account both the upper and the left samples in forming its prediction. The context model is depicted in Figure 4.2.

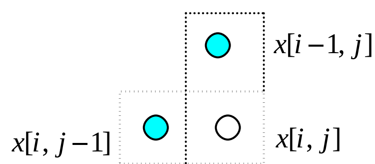


Figure 4.2 – Modified causal contest model

Looking into prediction theory [36], if we simplify the problem by restricting the predictor function to be linear, and the order of the predictor to be two in this model, the predictor $\hat{x}[i,j]$ is given by :

$$\hat{x}[i,j] = ax[i,j-1] + bx[i-1,j] \tag{4.1}$$

where a and b are linear predictor coefficients that can be chosen to minimize the prediction error s_d^2:

$$s_d^2 = E[(x[i,j] - \hat{x}[i,j])^2] \tag{4.2}$$

The coefficients, a and b, are calculated from the correlation matrix, **R**, and correlation vector, **r**, of the image being coded[1]:

$$\mathbf{A} = \mathbf{R}^{-1}\,\mathbf{r} \tag{4.3}$$

In this case

$$\mathbf{R} = \begin{bmatrix} f(1:1) & f(2:1) \\ f(1:2) & f(2:2) \end{bmatrix} \qquad \mathbf{A} = \begin{bmatrix} a \\ b \end{bmatrix} \qquad \mathbf{P} = \begin{bmatrix} f(0:1) \\ f(0:2) \end{bmatrix} \tag{4.4}$$

and

$$f(q:r) = f(k,l:m,n) = \sum_{i=L}^{U}\sum_{j=L}^{U} x(i-k,\, j-l)\,x(i-m,\, j-n) \tag{4.5}$$

where q and r are row wise scanning indices corresponding to two dimensional pixels:

$$q = I(k,l) \quad , \quad r = I(m,n) \tag{4.6}$$

L and U are limits of the region of support of the predictor. Here in the case of Figure 4.2, $L = 0$ and $U = 3$.

However, this technique needs to send the prediction coefficients to the decoder for each subband. In order to prevent the transmission of extra information, the prediction coefficients in this context model are set to have fixed values. Furthermore, the average of samples $x[i, j-1]$ and $x[i-1, j]$ makes a good choice for the predictor when all subbands are constrained to use the same coefficients. This choice is conservative since it assumes little in regards to the image structure. Averaging the two adjacent symbols is not the best solution for a context modeling, but it is also a reasonable way under the assumption that we know little about the distributions.

---

[1] See Maragos,Schafer and Mersereau [46] Section II for detail.

One approach to improve this predictor is to adjust it according to image features in the neighborhood of the sample. Theoretically, the adaptively selected predictor should have more precise approximation to the real value over the fixed predictor as mentioned above. There are various techniques for adapting the predictor without passing side information to the decoder. The next two adaptive predictors, Gradient Adjusted Prediction(GAP) and Median Edge Detection (MED), have been applied in recent research [45].

4.1.3 The GAP Predictor

The previous two fixed linear predictors have some disadvantages because their predictors are fixed for all symbols. The quality of the prediction can be improved by considering image gradients in the neighborhood of the pixel being predicted. The GAP technique in Xiaolin Wu's CALIC [58], described in Section 3.5, is a simple and efficient predictor which adapts prediction according to local gradients. It introduces two parameters d_h and d_v to represent horizontal and vertical gradients for each image sample during encoding. The data Wu used in his CALIC algorithm are 8-bit natural images, from which Wu obtained his edge orientation thresholds to be ± 80, ± 32 and ± 8 respectively [45]. Because the value (from zero to 64) ranges of the quantized SFS subbands are much smaller than the natural images, it is reasonable to assume that the thresholds should be reduced. The values ± 4, ± 2 and ± 1 were manually adjusted and determined by experiments, and the new thresholds perform better in quantized SFS subbands than the original ones given in Wu's paper. If the modeled sample $x[i, j]$ is at the edges of the image, the neighboring pixels of these edged symbols are unobtainable, and will be set to a default value, which is 32 for both natural-image (Barbara) and ultrasound-image (U1 and U7)[2] subbands. Because among the quantized subbands, 32 is the majority number when the quantizer size is 64, representing zero symbols.

The following flow chart, as shown in Figure 4.3, describes the adaptive prediction algorithm, where $\hat{p}$ stands for the predictor and $\hat{p}_t$ represents the temporal predictor. The uniformly quantized predictor, $Q(\hat{p})$, is treated as the context, *c*, for the conditional arithmetic coding.

The adaptive topology improves the prediction accuracy, especially for ultrasound images, which contain speckle. The improvement is approximately 85% to the above two predictors, Numerical comparisons can be found in Section 4.1.5. The alignment of the speckle shape can be known by estimating the gradient of each pixel inside the speckle. However, the adjusted predictor still can be improved upon, since the parameters and thresholds in this model are adaptive. Large prediction errors may still occur in images that are statistically different from those in the training set. In Section 4.1.5, the compression result of the conditional arithmetic coding using GAP is 2.85% and 2.44% respectively better than the previous two simple predictors. Its average prediction error is similar to the modified simple prediction model, but is 85% less than the simplest predictor.

---

[2] U1 is from the website (www.alitech.com) of ALI Technologies, which is a manufacturer of medical PACS ; U7 is obtained from the Radiology Department of Vancouver General Hospital. These two ultrasound images are used in Chiu's thesis[22], therefore, they are included here for comparison with BSF experimental results.

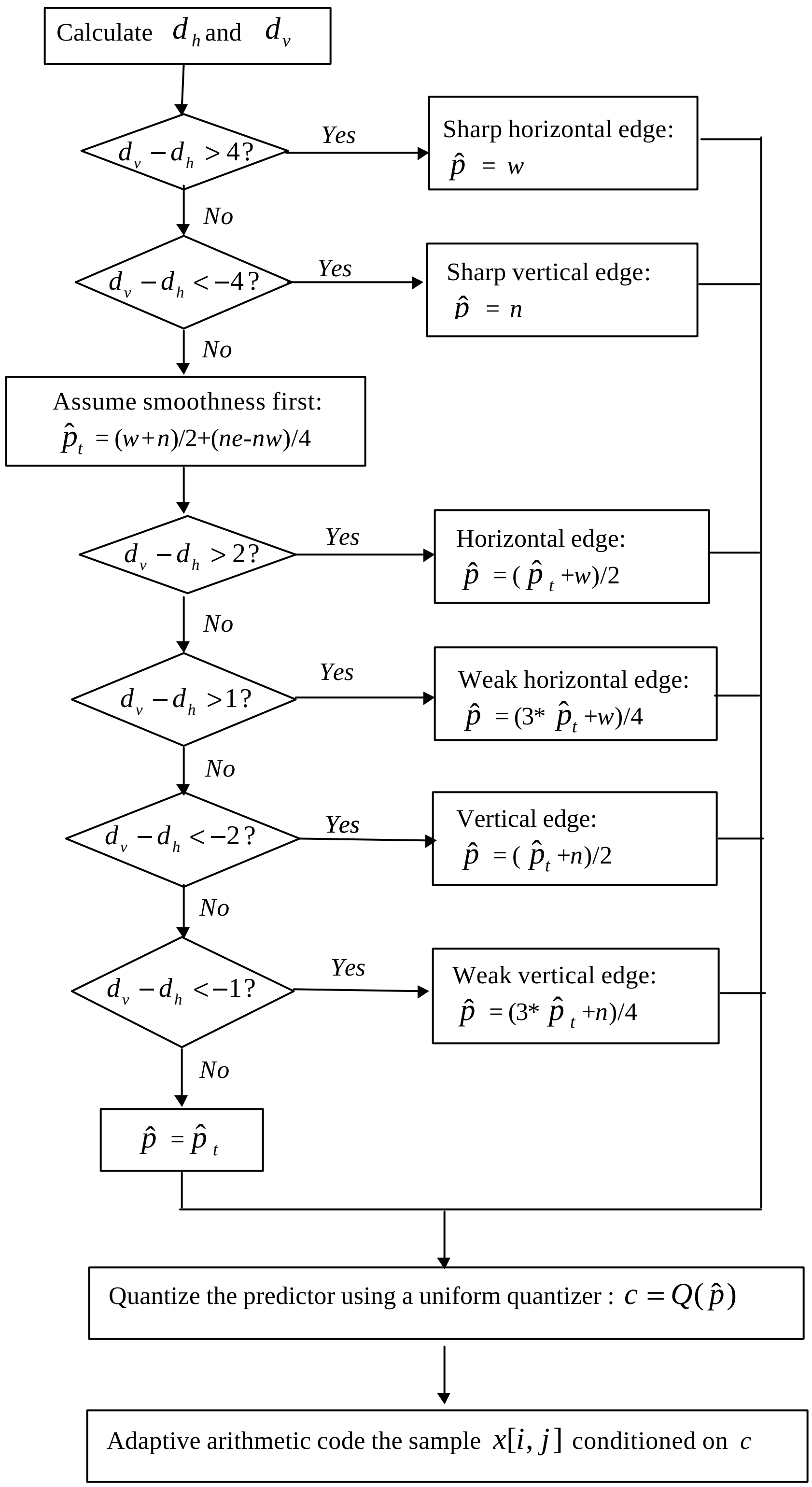


Figure 4.3 - Flow chart for GAP algorithm

4.1.4 The MED Predictor

The MED predictor is also known as MAP (Median Adaptive Predictor) and was first proposed by Martucci [48]. Hewlet Packard's proposal, LOCO-I (low-complexity lossless coder) [40], uses the MED predictor, which adapts in the presence of local edges. Memon and Wu [45] compare several predictors, such as the MED predictor, the GAP predictor, the Activity Level Classification Model (ALCM), and other predictors. Their data set is the ISO test images including RGB images, monochrome medical images and YUV video images. The method for producing the predictor is described in Figure 4.4 and Equation (4.7). The context is obtained from uniformly quantizing the predictor. They found that the MED predictor has the lowest average lossless compression rate, 4.204bpp, over the entire data set, comparing with 4.229bpp for GAP predictor and 4.259bpp on ALCM predictor.

Similar to GAP, MED also detects horizontal or vertical edges by examining the values of north(n), west(w) and north-west(nw) neighbors of the current sample $x[i, j]$ as illustrated in Figure 4.4.

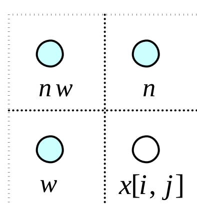


Figure 4.4 – The MED predictor

The north pixel is used as the predictor in case of vertical edge detection. The west pixel is applied in case of horizontal edge. Finally, if neither a vertical edge nor a horizontal edge is detected, an interpolation is used to compute the prediction value. Specifically, prediction is performed according to the following equations:

$$x[i,j]=\begin{cases}\min(n,w) & if \quad nw\geq \max(n,w)\\ \max(n,w) & if \quad nw\leq \min(n,w)\\ n+w-nw & otherwise\end{cases} \qquad 4.7$$

In an extensive evaluation, Memon and Sayood [43] compress the Y luminance components of a set of 576x720 color images from the JPEG test images to compare the entropy of prediction residuals using seven different prediction schemes, including the JPEG predictor [26], Hierarchical INTerpolation(HINT) [51], and the MED. The MED predictor was observed to give superior performance over most of these linear predictors. For example, the compression ratio of MED is 1.8% better than the JPEG algorithm and 4.3% better than the HINT scheme.

Like GAP, MED can also adapt to the speckle shapes during prediction. The horizontal or vertical orientation of the speckle is explored by the MED prediction. In our experiments of Section 4.1.5, MED has an average better compression result over the previous three predictors at 4.39%, 3.97%, and 1.50%, respectively. Its prediction precision is much better than the simplest predictor and similar to SP1 and SP2. As shall be seen, MED is chosen as the causal context model in our proposed algorithm due to observation for compression as well as prediction.

4.1.5 Experiments to Compare Causal Context Models

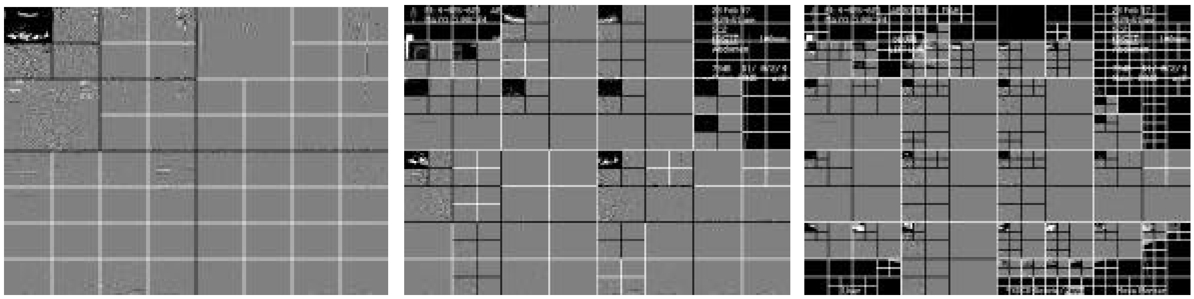
Experimental performance between different causal context models is compared in this section. There are four predictors:

- SP1, the Simple Predictor 1 discussed in Section 4.1.1.

- SP2, the Simple Predictor 2 which is described in Section 4.1.2.
- The GAP predictor that can be found in Section 4.1.3.
- The MED predictor presented in the previous section.

In order to generate experimental subbands for the proposed research, 54 quantized SFS subbands are obtained from both the natural and ultrasound images. The SFS is implemented at different decomposition levels to obtain different size and statistically different subbands. Figure 4.5 illustrates three space-frequency presentations of U1 at different decomposition levels, with the target bit rate is 0.5bpp. Subband images 1 to 25 of Table A.1, Appendix A[3] are manually selected from similar decomposition levels from the Barbara image while subband images 26 to 54 are selected from the decompositions shown in Figure 4.5. The selection of these experimental subbands are done under the consideration of including various kind of subbands in the test. We have different images, different decomposition levels, different band sizes, different frequencies and differently partitioned subbands. Two to three subbands are chosen randomly among each kind. The values of sample subbands have all been quantized during the optimization of SFS segmentation. Detailed descriptions of the quantizers can be found in Chiu's thesis [22]. The largest alphabet size of most quantizer sets is 64. The step size for each subband will be selected from a fixed set according to the rate-distortion cost and bit allocation when given the target bit rate.

[3] The names, sizes and other attributes of these 54 experimental subbands are listed in Table A.1 of Appendix A.



level=3 level=4 level=5

Figure 4.5 – SFS presentations of U1

Working on the experimental subbands, the precision of the above four predictors was estimated by calculating the mean square error between the predicted and the original images. The performance of the prediction schemes are compared through conditional arithmetic coding rates using different contexts. The context for each subband coefficient is obtained from quantizing its predictor as we discussed in the previous sections. The number of quantizer outputs range from one to ten. The context size, equal to the quantization output number, is optimized during encoding each subband from one to ten for all of the four context models. Experimental evidence shows that there is no optimal context size greater than ten in the compression schemes, therefore ten is chosen be the largest context size for the entropy coders. Having the actual maximum and minimum value of each subband, $x_{\max}$ and $x_{\min}$, the step size, s, of an uniform quantizer is obtained from these values and the output number, nC.

$$s = \frac{x_{\max} - x_{\min}}{nC} \qquad 4.8$$

It is not the only way to generate uniform quantizers, and it is not necessary to apply a uniform quantizer. However in a conservative consideration assuming we know little about the data structure, it is an intuitive and simple way.

The compression results are listed in Table 2 of Appendix A. The lowest bit rates among these causal context models are shown in bold words for each subband. Figure 4.6 and 4.7 give a graphical view of the comparison between these four predictors on the entire set of 54 subband images, the result for basic arithmetic coding is also included in Figure 4.7 as a reference labeled as “BA”. In these figures, the subband sizes decrease from left to right for each image, as listed in Table 4.1.

Table 4.1 – Subband image sizes

Image type	**Subband images**	**size**
Natural	img1 – 4	128x128
	img5 - 15	64x64
	img16 – 23	32x32
	img24 – 25	16x16
ultrasound	img26 – 29	120x160
	img30 – 38	60x80
	img39 – 45	30x40
	img46- 54	15x20

The performance of the basic arithmetic coder shown in Figure 4.7 is not as good as any of the four context-based coders. This proves that there are dependencies among the quantized SFS subbands, so that the conditional arithmetic coding conditioned on the context model achieves a better compression result than basic arithmetic coding.

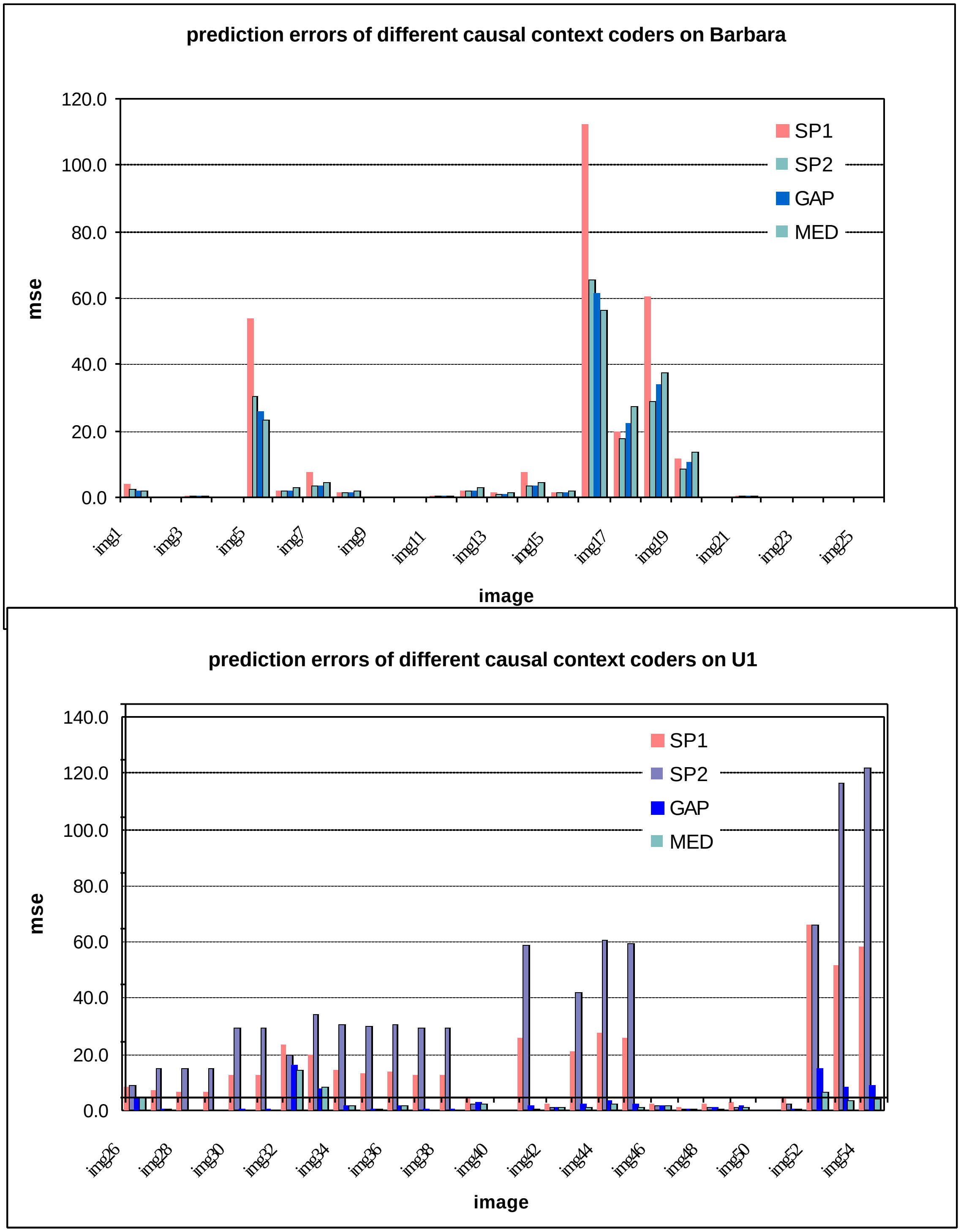
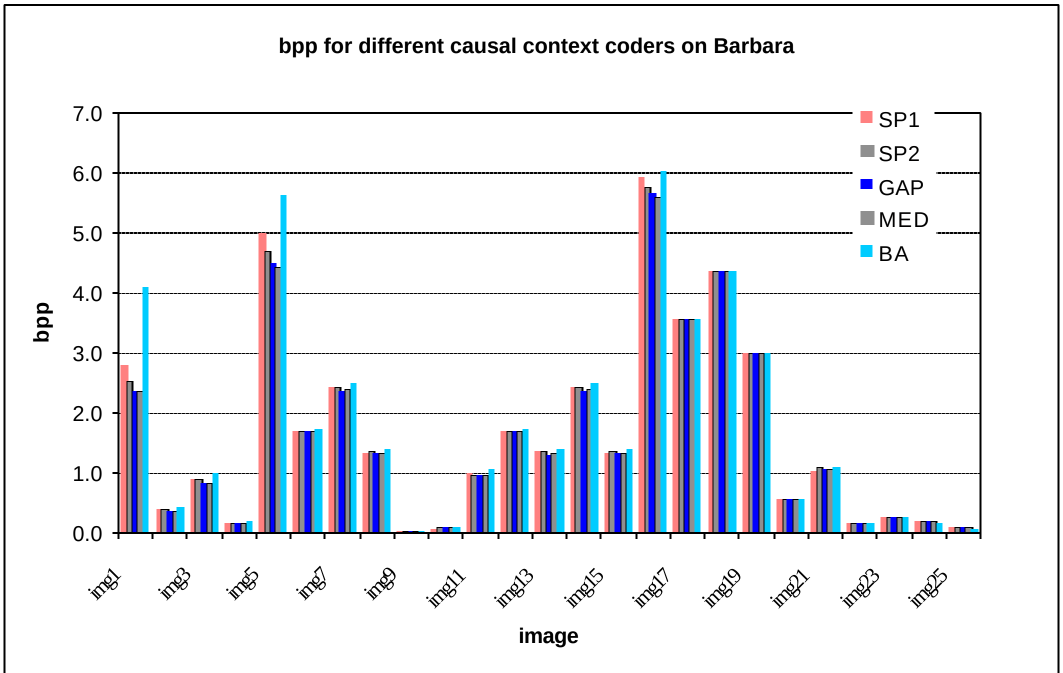


Figure 4.6 – Predictor errors of the four causal context coders



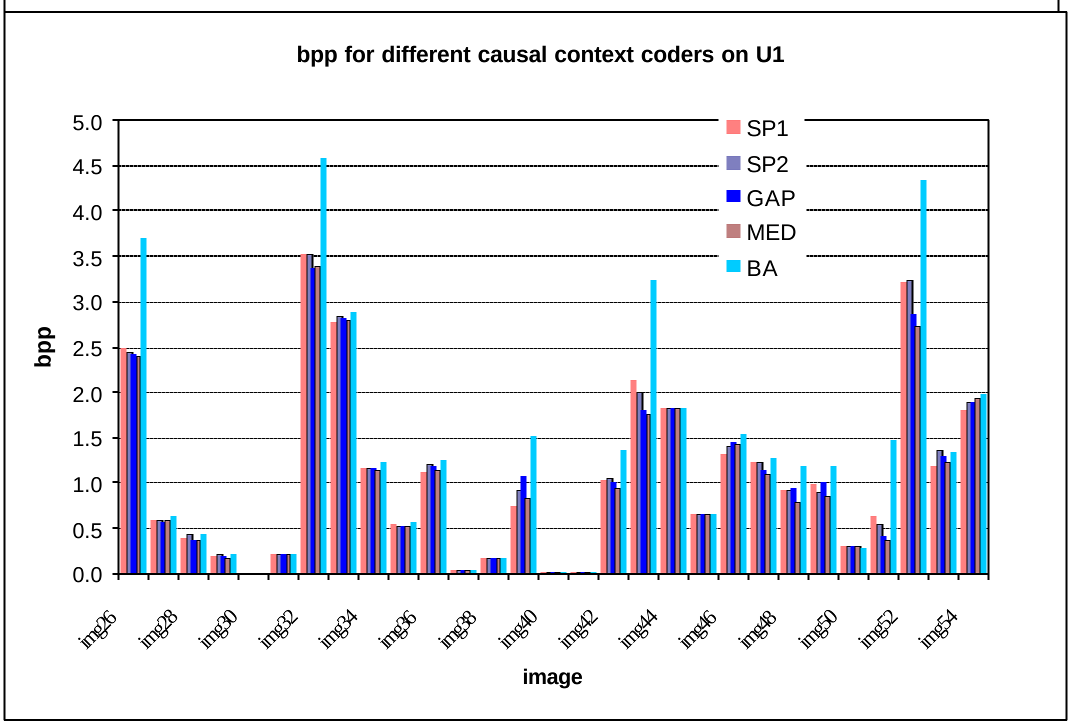


Figure 4.7 - Compression performance for the four causal context coders

Another evidence from these plots is that the prediction errors in Figure 4.6 follow the same patterns as the compression rate of Figure 4.7, which implies that the better the prediction precision, the more efficient the compression. This emphasizes our intention to find a good predictor model for the proposed context-based entropy coder.

In Figure 4.6 and 4.7, img1,5,16,32 and 52 dominate the performance due to their higher bit rates and prediction MSE compared with the other subband images. Therefore, it is not clear to simply compare the four coders from the bar graphs of the entire sample subband images. Detailed statistical analysis on the compression performance and the prediction precision are summarized in Table 4.2.

Table 4.2 – Performance comparison for the four causal models

	No. of subbands for best compression result		Average predictor error (MSE)	
	Natural image	Ultrasound image	Natural image	Ultrasound image
SP1	2	6	11.5342	15.9264
SP2	0	0	6.8532	29.5019
GAP	9	3	6.9723	3.0543
MED	8	11	7.3111	2.1833

SP1 is the simplest predictor for a causal context model, but its compression performance and prediction precision is not as good compared to GAP and MED. Neither could SP2 obtain good compression results, although its MSE values in the natural-image subbands is close to the best obtained using the four models. The poor compression performance of SP2 can be explained by the simple context modeling, which is the average of the nearest two causal adjacent samples for each pixel. The predictor coefficients should be optimized for the class of images being coded for improved performance. The poor prediction of SP2 on ultrasound images also explains why it is not competitive with the other coders for ultrasound images. The experimental results suggest that GAP and MED work better, and they tend to give better prediction on ultrasound images. Furthermore, the computational complexities of GAP or MED are not very high (algorithms are described in Section 4.1.3 and 4.1.4). They could both be considered as representatives of causal context models for more experiments. However, looking into the experimental results carefully, we find that most of GAP's best compression performance appears in the large subbands of natural images. When applied to medical images or smaller size subbands, it does not work as well as MED, which has the best performance among the four coders on over half of the medical subband images or small-sized[4] subband images. In practice, small subbands are often obtained by the SFS algorithm. Therefore, the GAP predictor will not be chosen for the proposed research. The following experimental result confirms this choice: all of the 128x128 (Barb) subband and half of the 160x120 (U1) subbands' best compression results are obtained from GAP scheme, whereas in smaller subbands, GAP does not appear to work well.

---

[4] When the subband size is smaller than or equal to 20x15(ultrasound image) or 16x16(natural image), they belong to small-sized subband. Details see Section 5.2.1.

Based on the above discussion, MED is selected to be the best appropriate causal context model on the BSF subbands of ultrasound image. Further experiments will be performed in Chapter 5.

4.2 Non-Causal Context Models

Causal context models cannot spatially enclose the modeled sample, because only the left and upper neighbors are used in modeling and predicting. Many image features, such as the intensity gradient, edge orientation, and textures can be better modeled in a completely enclosing context, which leads to non-causal context modeling.

The Context Selection, Quantization and Modeling (CSQM) technique, which Xiaolin Wu used in his lossless compression of continuous-tone images [59] is a non-causal context modeling scheme. Generally speaking, the distribution of pixel values in the residual image[5] can be closely approximated by a zero mean Laplacian distribution [3,30]. The probability density function of such a distribution is given by

$$p(x) = (a/2)e^{-a|x|} \qquad 4.9$$

where $a^2 s^2 = 2$, with s^2 being the variance. Wu proposed a Delta Pulse Code Modulation(DPCM) compression algorithm based on the above hypothesis and showed that it gives good performance for natural 8-bit images when compared with other well-known lossless compression

[5] The image generated from prediction errors.

algorithms, such as JPEG method [26], HINT algorithm [51], Fast Efficient Lossless Image Coding System (FELICS) [1], and the Universal Context Modeling (UCM) algorithm[41]. The CSQM lossless compression scheme will be used as one of our non-causal context models. Detailed descriptions of the context formation is presented in Section 3.5.

Encoding the prediction residuals on the basis of the CSQM context modeling seems to be quite straightforward since we have efficient coding techniques that encode at a rate very close to the zeroth-order entropy. Unfortunately, Memon and Sayood found that this property is not always true for wavelet subband images in their research [43], and their situation is similar to that of the quantized subband images used in our experiments. They observed that even after applying the most sophisticated prediction techniques, the residual image has ample structure. In fact, the distribution of the residual values is sharply peaked at zero in the more uniform areas of the original image. However, it is not always the same in more active regions where the residual value is different from zero with high probability. The quantized wavelet subband images belong to this case. Simply coding the subband image symbols instead of their prediction residuals seems to be suitable and practical in our research.

A multi-resolution representation of an image, with each layer in the hierarchy representing the entire image but at different spatial resolutions, has the advantage that the receiver can recover the image level by level. After reconstructing each level, the receiver can obtain an approximation of the entire image by interpolating unknown pixels. This multi-level coding technique, which forms a non-causal context model is found to be useful in progressive transmission schemes [39]. Taubman's multi-rate coding scheme (Layered zero coding and layered PCM/DPCM coding) forms one part of JPEG2000 technique, and his ideas we used as the basis for one of our non-causal context models for comparison purposes.

4.2.1 Context Selection, Quantization, and Modeling Schemes (CSQM)

Referring to Section 3.4, we know that the prediction context in this CSQM algorithm forms a complete spatially enclosed 360° region surrounding the current pixel. The context is modeled by the neighboring pixels from any direction around the currently coded symbol. The causal neighboring pixels are easily obtained for both the encoder and the decoder. For those non-causal contexts, those at the right and bottom sides of the symbol, we can apply an alternative way to obtain these values from the previously coded layer or sub-images at the same position. Wu applied a sub-image to form a non-causal context model in his three-pass algorithm [59], as described in Section 3.5. In this section we discuss the algorithms of CSQM and DPCM on CSQM, which are both from Wu's research.

In CSQM, Wu applied a dynamic programming technique to partition the "error strength discriminant", Δ, of each symbol in the image to obtain the context. Under the circumstance that we know little about the structure of the quantized subband images, also in order to simplify the problem, we instead apply uniform quantizers to specify the contexts. The results show that this method works only 3.8% worse than Wu's method when coding the same experiment images. After estimating contexts, we conditionally entropy code the coefficients using different adaptive arithmetic coders according to their contexts. The number of contexts is eight, as explained in Section 3.5.

Now let's look into the algorithm of DPCM on CSQM with adaptive error modeling. It is discussed here for comparison with the simple CSQM algorithm described above and the multi-rate scheme presented in Section 4.2.2.

Like the method of context modeling presented above, in Wu's DPCM algorithm [59], the context is also determined using both the quantized error strength discriminant d, where $d = Q(\Delta)$, and the spatial texture patterns t; eight contexts are again used. What is more in this algorithm is that the expected value of the prediction error $\bar{\varepsilon}(d,t)$, conditioned on each compound context (d,t), is estimated and added to the predictor $\hat{x}$, which is obtained from the three interlaced passes, to generate an improved context-sensitive predictor $\tilde{x} = \hat{x} + \bar{e}(d,t)$. The predictor error e is obtained from $e = x - \tilde{x}$, and is entropy coded using a conditional arithmetic coder based on the context $C(d,t)$. This adaptive context-based error modeling means that most of the prediction bias of the linear predictor $\hat{x}$ can be corrected in the given compound context, since the average prediction error $\bar{\varepsilon}(d,t)$ is added to the predictor $\tilde{x}$.

In addition, the sign of the conditional error means $\bar{e}(d,t)$ can also be utilized to sharpen the conditional probabilities $p[e \mid Q(\Delta) = d]$, and reduce the underlying conditional entropy of the predictor error e, as explained below. From the experiments, Wu found that $p_+[e \mid d]$ and $p_-[e \mid d]$ are approximately mirror images of each other, where

$$p_+[e \mid d] = p[e \mid d, \bar{e}(d,t) \geq 0] \qquad 4.10(a)$$

and the similarly with $p_{-}[e \mid d]$:

$$p_{-}[e \mid d] = p[e \mid d, \bar{e}(d,t) < 0] \qquad 4.10(b)$$

Therefore, Wu flipped $p_{-}[e \mid d]$ around the origin and imposed the flipped $p_{-}[e \mid d]$, namely $p_{-}[-e \mid d]$, onto $p_{+}[e \mid d]$ to create an approximation of $p[e \mid Q(\Delta) = d]$, which is presented by $\hat{p}[e \mid d]$. The technique of error sign flipping techniques applied here improves the coding efficiency because the context modeling of errors captures the statistical redundancy in the sign of e. Figure 4.8 gives the main steps for this algorithm.

As we discussed above, although Wu presents good results from his experiments on natural 8-bit images, this DPCM algorithm on CSQM context modeling may not show its power on our quantized subband, because the statistical structure of natural images and the quantized SFS subband images are quite different, and the values in the quantized subbands are usually small, maximum value of 64 for most subbands, compared with natural 8-bit images, which have the maximum value of 256. The key difference between the above two algorithms is that the CSQM scheme encodes image symbols, while the method of DPCM on CSQM encodes the prediction errors of the symbols. The experiments in Section 4.2.3 compare results of these two algorithms and the multi-resolution prediction model discussed in the next section.

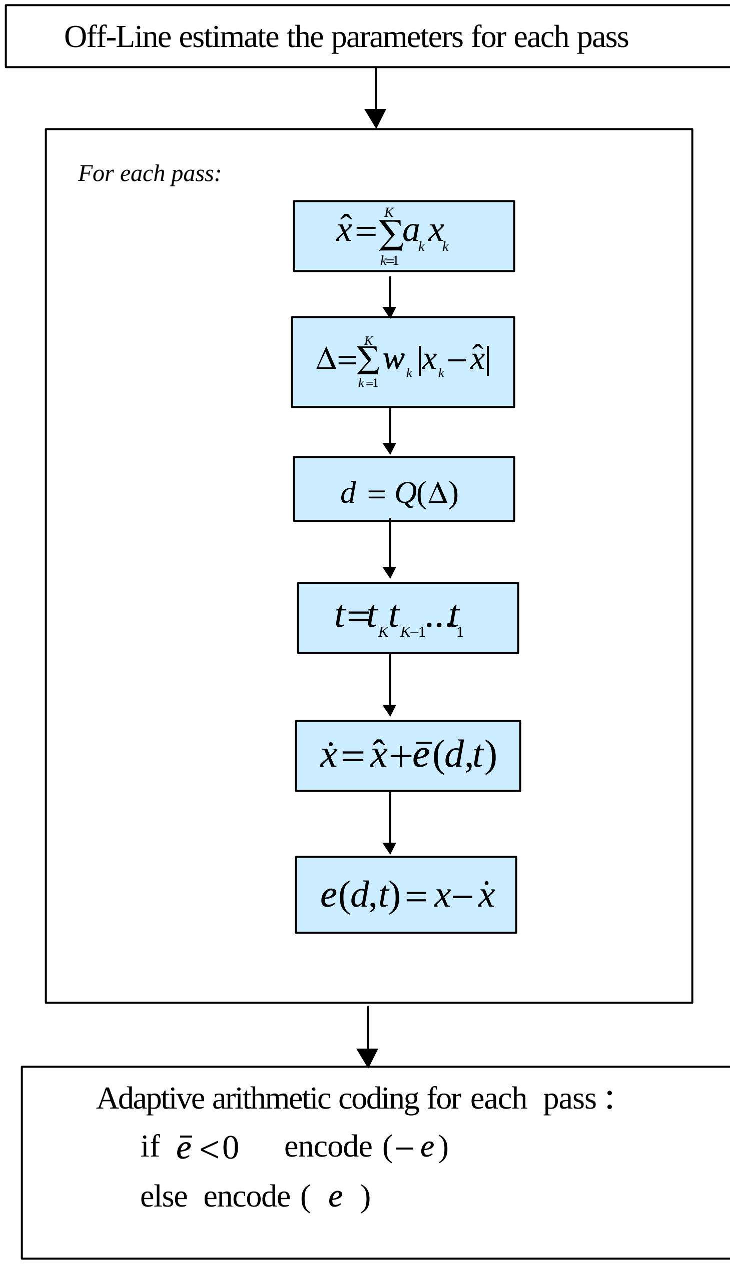


Figure 4.8 – Flow chart for DPCM with CSQM algorithm

4.2.2 Multi-resolution Prediction Models

David Taubman and Avideh Zakhor have proposed a technique for multi-rate 3-D subband Coding of Video [18] which includes layered Zero Coding and layered PCM/DPCM.

In our research, we modified his multi-resolution algorithm and applied it to our quantized SFS subband images. From the description in Section 3.5, we know that the multi-layered quantization in Taubman's scheme is applied to the subbands using a bit allocation strategy, LZC and Layered PCM/DPCM is then applied on each quantization layer. However, in our proposed experiments, in order to work under a fair condition with the other entropy coders, the experimental subbands have already been quantized from the SFS optimization based upon rate-distortion criteria. Therefore, we design a set of quantizers to quantize our particular subband images and produce multiple resolution layers for LZC and Layered PCM in our multi-resolution prediction model. Because the quantized SFS subbands have only integer values, all the step sizes of our designed quantizers are integers. Furthermore, the finest quantization step size is set to be one in order to guarantee a lossless compression at the end. Taubman chose nine quantization layers in his research [18] for both luminance and chrominance subbands. In our quantized SFS subband, the index values are not so large, therefore, we use five quantization layers in our experiments. Experimental evidence shows that five are better than nine for our particularly quantized SFS subband, the following table list all the quantization layer N, and step sizes Δ_n for both Taubman's and our quantizers, where T_{lum} and T_{chr} represent quantizers in Taubmans' luminance subband and chrominance subband respectively, and pr. means our proposed quantizer.

Table4.3 – Quantizer comparisons

	N	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5	Δ_6	Δ_7	Δ_8	Δ_9
T_{lum}	9	256	128	64	32	16	8	4	2	1
T_{chr}	9	200	100	50	25	12.5	6.25	3.125	1.5625	0.78125
pr.	5	16	8	4	2	1	N/A	N/A	N/A	N/A

In Taubman's layered zero coding, the context size is 66 as explained below. Since the value for the binary sequence $s_n[i, j]$ is either zero or one, the largest possible value for the local conditioning information $k'_n[i, j]$ is 63 according to Equation (3.12). Furthermore, the largest possible value for the wide conditioning information, $B_n^4[i, j]$ and $B_n^8[i, j]$, is one. Therefore, from (3.15), the maximum value for the conditioning terms $k_n[i, j]$ is 65, which means that $s_n[i, j]$ has 66 context states in total. Initially in our research, we optimized the context size over the range one to 66, but we finally fixed it to be 66 instead of being adaptive during coding, as a tradeoff of the compression performance and the computational complexity. A detailed discussion is presented in Section 4.2.3. For those non-zero symbols, Layered PCM is applied on each layer. The context number for the n^{th} quantization layer L_n is $2M_n$ where M_n is the maximum output number of quantizer Q_n. Detailed description can be found in Section 3.5. The compression results of these non-causal context models described in the following sections show that the modified Taubman's multi-rate algorithm performs about 14% and 16% better than the CSQM and DPCM_CSQM coders respectively. Technically speaking, the multi-rate coder we designed is not exactly the same as that of Taubman's [18], we use five quantization layers instead of nine in their algorithm, and we apply PCM on LL subband instead of DPCM for each layer. But the core of our coder is still multi-resolution in the sense that we used five quantization layers from coarse to fine in each subband. The non-causal context for layer L_i is derived from the L_{i-1}. Although the quantizers and the quantization layers are no longer the same as those used in Taubman's algorithm because of different subband images, the result of the compression given in the next section shows this context modeling method is still an effective entropy coder when comparing with other algorithms, such as the basic arithmetic coder.

4.2.3 Experiments to Compare Non-Causal Context Models

In this section, we still use the same 54 experimental subbands as described in Section 4.1.5, and compare the compression performances of the following three non-causal context models:

1. The CSQM context modeling coder, which encodes the subband coefficients.
2. The DPCM on CSQM algorithm, which encodes the prediction errors of the subband coefficients.
3. The multi-resolution coding scheme, which is described in Section 5.2.2.

Experiments have been designed to study these models, and the most appropriate coder will be selected using compression performance.

Experiment 4.2.1 : Research on the distortion problem for the CSQM context modeling

Recall in Section 3.5, from the definition of $m[i, j]$ in equation (3.4), we found that there would be a distortion of 0.5 on $m[i, j]$ if the sum of $x[i, j] + x[2i+1, 2j+1]$ is an odd integer. In the second pass of the prediction, $x[2i+1, 2j+1]$ is calculated from $x[2i+1, 2j+1] = 2m[i, j] - x[2i, 2j]$, therefore, a truncation error of one will be generated. Assuming that in the worst case, all of the pixels of $x[2i+1, 2j+1]$ have an error of one, leading to a MSE of 0.25 from $[(height/2)\times(width/2)]/(height\times width)$. Hence the PSNR for the reconstruction image is :

$$10\log\frac{255^2}{MSE} = 10\log\frac{255\times 255}{0.25} = 54.15 \qquad 4.11$$

This minimum and spatially isolated error can be ignored and CSQM is actually a near lossless compression technique. But in our algorithm, the values in each subband to be entropy encoded are indices after quantization. A difference of one in this index implies much more difference for actual value after decoding. If this truncation error exists, it makes no sense to estimate the rate-distortion for optimization afterwards. The problem can be dealt with by encoding the truncation error in an extra file, then transmitting it to the decoder through an adaptive arithmetic coder, leading to higher bit rate.

We work on two schemes that result in two rates: $r1$ is obtained when the truncation error is ignored, which leads to an extra (about 132%) distortion, besides the quantization error. Rate $r2$ is achieved after transmitting an extra file containing the truncation errors, coded by the basic arithmetic coder, to the decoder to guarantee a lossless compression. The extra file containing the truncation errors is coded by arithmetic coding. Table 3 and Table 4 in Appendix A list the compression results on all of the 54 subband images from CSQM context modeling coder and DPCM on CSQM algorithm respectively.

The two rates are compared. Rate $r1$ is lower than $r2$, and the average percentage of the difference between $r1$ and $r2$ is 24.0% for CSQM context modeling coder and 15% for DPCM on CSQM algorithm. This difference is significant, and the extra bits can not be ignored.

Experiment 4.2.2 : Context number of non-causal models

From Section 4.2.2, we learn that the context size for the multi-rate zero coding is 66. Furthermore, we also know that if the subband is small enough, more context states does not mean better compression. Therefore, the context size should be swept from one to 66 to find an appropriate value. However, the optimization in the algorithm takes a long time to execute, and the running time will be 66 times more than that of the non-optimizing algorithm. In order to find out the difference between the two coders (the coder with optimal context number and the one with a fixed context number), compression performance is evaluated and results[6] are illustrated in Figure 4.9, where r_opt means the rate with optimized context sizes and r_fix represent the rate with fixed context size. To estimate the context c, the temporary context c_{66}, assuming that the context size is 66, is calculated first, then c_{66} is multiplied by a factor which is equal to $c_s/66$, where c_s is the desired

[6] Detail results are listed in Table 5 of Appendix A.

context size. There are also other methods to generate context sizes except numerically merging adjacent contexts, but our method is a simple way to reduce the context number and we can obtain any desired context number from one to 66. This method is also applied in the decoder. The reconstructed image shows the correctness of this scheme.

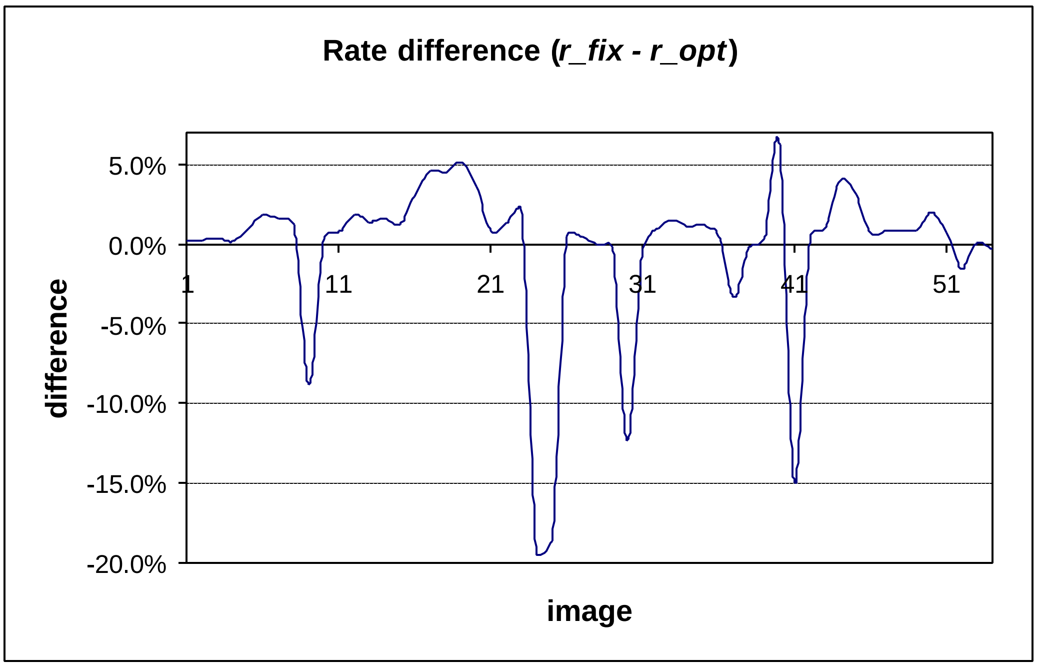


Figure 4.9 – Difference between rate r_opt and r_fix

The average difference between two rates is 0.3%, so we see that optimizing the context size does not have any advantage over fixing the context number. Furthermore, the rate from the optimal context number algorithm is even larger than the fixed one on average. It is reasonable to have such result, because some small subbands have only a single non-zero value, such as img41 and the other three images which have a much smaller r_fix in Figure 4.9, need more bits to transmit the optimal number of contexts, leading to a much higher

r_opt. In addition, medical images are much more likely to contain such subbands. On a natural image (Barbara), nearly all of the r_opt are smaller than r_fix, except the two smallest subbands (16x16). However, for ultrasound image (U1), nine out of twenty-nine subbands have smaller r_fix. Considering the average bit rate and the computation complexity, r_fix is chosen to represent the multi-resolution coder, and 66 is fixed to be the context number applied in the algorithm.

In Taubman's algorithm [18], layered PCM is applied in high frequency subbands while layered DPCM is applied in the low frequency subband. Layered DPCM does not work well on the low frequency subband in our SFS algorithm, since the low frequency subband is quantized to a small number of levels. The prediction error is thus similar in magnitude to the original quantized coefficients. As a result, the coding gain is very limited. Therefore, layered PCM is applied on all of the subbands.

Experiment 4.2.3 : Performance comparison between the three non-causal context models

In this set of experiments, we compare the three non-causal context coders. We apply them to entropy code the 54 example subbands as explained in Section 4.1.5. The context size for CSQM and DPCM on CSQM algorithms are eight, and for the multi-resolution scheme, the context size is 66, as explained in the previous sections. Compression results are listed in Table 6 of Appendix 1, from which detailed comparisons are presented in Table 4.3 and bar graphs are depicted in Figure 4.10, where the basic arithmetic coder (BA) is also included for comparison. In the CSQM and DPCM on CSQM algorithms, compression rate is $r2$ as described in experiment 4.2.1.

The DPCM technique appears to be efficient for low frequency subbands in Taubman's algorithm. However, this kind of algorithm does not work for our SFS subband images, which consist of mainly high frequency subbands. From the bar graphs, the CSQM modeling techniques can not compete with the multi-rate coding method. One possible reason is that the sub-image coded in the first pass, which is used for future prediction, is comparatively large when the subband size is small, and the credit of the 360^0 context modeling can not make up the loss in compression rate. The extra "truncation" bits discussed in experiment 4.2.1 is another possible reason. Therefore, CSQM is not a good choice among the non-causal context models for our particular experiments. From Table 4.4, the basic arithmetic coder works better than other entropy coders , especially for small-sized subbands as shown in Figure 4.10. This emphasizes the necessity of coding the small-sized subbands using the basic entropy coder. Section 5.2.1 gives detail discussion.

Table 4.4 – Comparison between non-causal context models

	No. of subbands for best compression result	
	Natural image	Ultrasound image
CSQM	0	0
DPCM on CSQM	3	3
Multi-resolution	12	11
BA	10	15

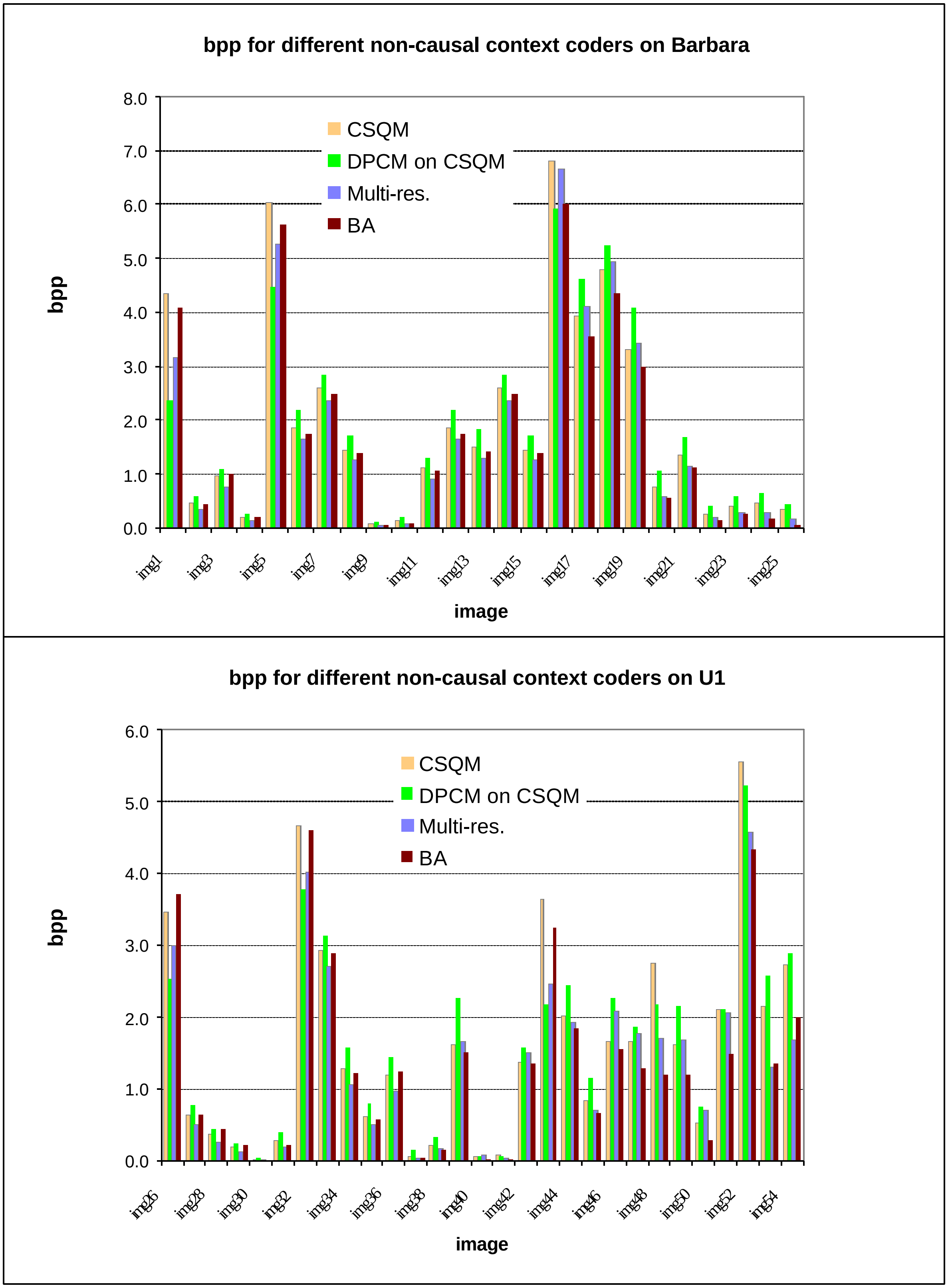


Figure 4.10 – Compression performance for non-causal context coders

It is quite clear from the result that the multi-resolution algorithm outperforms the other two non-causal context models on both natural and ultrasound images. Therefore it is the selected as the entropy coder to represent the non-causal model on the SFS subbands for further comparison with the selected entropy coder for the causal model. The PSNR results for the entire image will be presented in Chapter 5.

4.3 Optimal Number of Context States

The number of context states is an interesting point in context modeling that directly influences the compression efficiency for a context based entropy coder.

In adaptive arithmetic coding, both the encoder and decoder learn the probability of each coded symbol and then generate distribution histograms while encoding and decoding. For conditional arithmetic coding, several histograms are applied to achieve higher compression efficiency as we discussed in Chapter 2. Every symbol is categorized into a different histogram associated with each context. The number of histograms is the same as the number of context states. Having selected the number of context states to give the lowest compression rate for the image, the conditional entropy coder can achieve a better compressed binary file through better matching of probability density functions.

Apparently, the smaller the size of a subband, the smaller the number of the context states it can support. The reason is very intuitive: when the subband size is small, the number of symbols available for learning are limited. So a large number of contexts means that a large number of histograms have been unable to learn the statistics well. In comparison, a smaller number of contexts not only prevents the dilation problem, but also reduces the time and space complexities as well. When the image size is large enough, the proper number of contexts produces better matching histograms for better compression results.

Optimizing the number of the context states for each subband is combined in our proposed entropy coder. The number ranges from one to ten for the causal context coder. For the non-causal context coder, fixed context sizes are applied. The CSQM model has eight context states, and the multi-resolution model has a context size of 66. These selections have been discussed in the previous sections and experimental performance of these coders when incorporated with SFS partition is presented in Chapter 5.

4.4 Summary

In this Chapter, we have presented two types context-based entropy coders: one type using causal prediction-based context models and the other using non-causal context model. Each model contains several coders for comparison.

In the causal prediction-based context model, we have discussed the simplest causal model, the modified causal model, the GAP and the MED. Based on our concept proof experiments, we have chosen MED to present causal context models due to its outperformed compression ratio and prediction precision. Using multiple quantization layers, our multi-resolution scheme is selected as the best coder among the non-causal context models (the CSQM, the DPCM on CSQM and the multi-resolution algorithms).

Further experiments are designed to study these selected context–based entropy coders when combined with the space-frequency segmentation algorithm. Detail results are shown in Chapter 5.

Chapter 5

Improvement on SFS

In this chapter, various context models are investigated. Experiments are devised to demonstrate the effectiveness of context models when applied to the SFS subbands of ultrasound images. Since the space-frequency segmentation algorithm is greatly influenced by the entropy coder that it employs, replacing the conventional arithmetic coder with the better performing context based entropy coder will further enhance the performance of the SFS algorithm.

5.1 Experiment Designs

The experiments designed in this section accomplish two purposes. First, we compare causal and non-causal context models applied to quantized SFS subbands. We then use the above results to select the most appropriate context model for incorporation into the entropy coder of the space-frequency segmentation algorithm. The second set of experiments evaluates the performance of the modified SFS algorithm. We manually select 54 representative subbands from various decomposition levels from both the natural image (Barbara) and the ultrasound image (U1) to use as the basis for our experiments. The selection of these subbands was rationalized in Section 4.1.5.

To ensure valid results, we implemented independent encoders and decoders, with bit rates determined from the resulting file sizes. The reconstructed images verify the correctness of our implementation.

Experiment 1: different context models

Selected causal and non-causal context models from Chapter 4 are used to drive arithmetic coders that operate on the quantized SFS subbands. The basic arithmetic coder without context modeling is used as a basis for comparison.

Experiment 2: space-frequency segmentation.

The SFS decomposition in Chiu's method is controlled by a rate-distortion function, with the rate evaluated using either a basic arithmetic coder or a stack-run coder. When we add context modeling to the arithmetic coder instead of the basic arithmetic coder, the lower bit rates affect the rate-distortion tradeoff and result in different partitioning decisions. Experiments are done with full ultrasound images, "ultrasound-only" images and natural images to determine if there is a systematic effect on the partitioning and to quantify the PSNR improvement at a given target rate. We compare the PSNR performance with that obtained using SFS with "basic arithmetic coding" (see Section 3.3), and the SPIHT (Section 2.1.3).

5.2 Experimental Results

The following experiments are designed to choose an appropriate entropy coder for combining with the space-frequency segmentation in the coding of ultrasound images. Our selection is made from the space of possible context models studied in Chapter 4, where the causal and non-causal coders are selected from the concept proof experiments.

5.2.1 Performance of Different Entropy Coders

In studying the causal context-based entropy coders, the number of contexts is varied during coding, ranging from one to ten, as explained in Section 4.1.5. When the context number is chosen to be one, it looks like the basic arithmetic coder, but it has to transmit a header file containing the optimized context number and the maximum and minimum subband coefficient values for calculating the step size of the quantizer. We cannot only transmit the step size instead of both the maximum and minimum values due to the unsymmetrical property of the quantizer. Therefore, instead of a subset of the proposed coder, the basic arithmetic coder is actually better than the above coder with one context due to the side information issue. In the experiments, the number that achieves the lowest compression rate is selected as the context size for that subband. The number of context models has been effectively put inside the optimization loop in the experimented coders, thus increasing the computational complexity by a factor of about ten. This extra complexity can be removed by using a fixed context number for subbands of specific sizes at only a small performance penalty (See Section 5.2.2). In the non-causal case, the context size for the multi-layer zero-coding is fixed to 66 as a tradeoff between the computational complexity and the compression ratio as discussed in Section 4.2.2. The cumulative frequency histograms in the arithmetic coding for our proposed causal and non-causal context coders are uniformly initialized to have one count for each symbol.

The experimental results show that compression performance is different in different sized subbands. Therefore, we divide the subbands into three groups according to the subband sizes shown in Table 5.1.

Table 5.1 – Different subband size

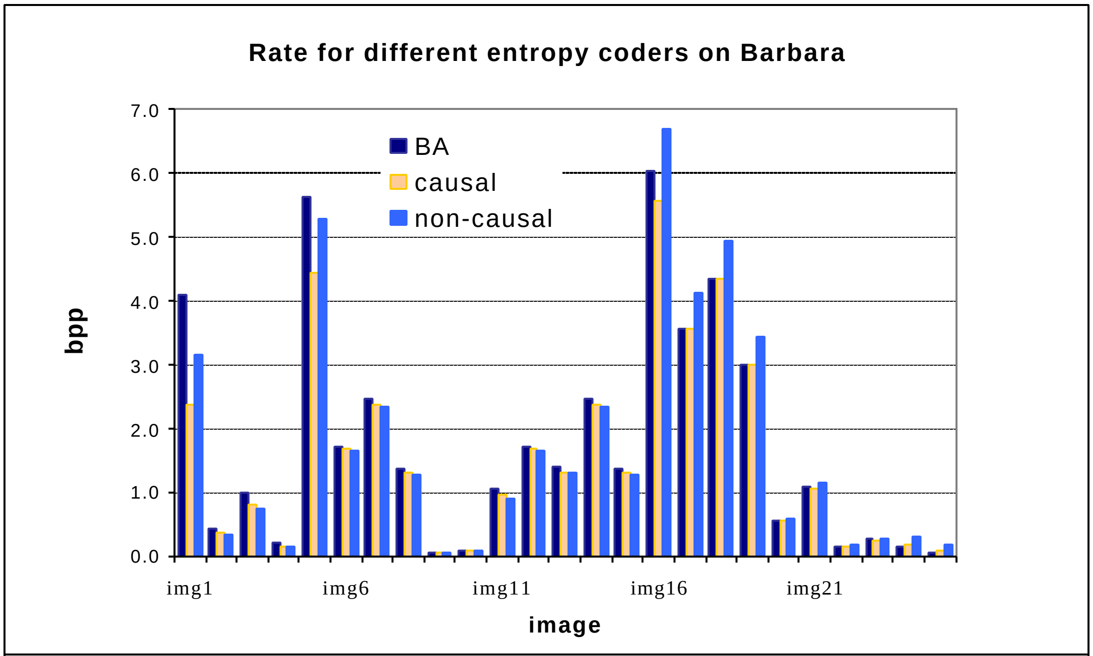
Subband type	**Natural image**	**Ultrasound image**
Small-size	smaller than or equal to 16x16	smaller than or equal to 20x15
Medium-size	between 32x32 and 64x64	between 40x30 and 80x60
Large-size	larger than 64x64	larger than 80x60

The summary of the compression comparisons for each subband group is given in Table 5.2, although we have that comparison in Figure 4.7 and 4.10. The results are close, therefore the small difference is hard to seen clearly.

Table 5.2 – Performance comparison between different entropy coders

	No. of subbands for best compression result					
	Natural image			Ultrasound image		
	Small size	Medium size	Large size	Small size	Medium size	Large size
Basic arith.	2	6	0	1	7	0
Causal	0	5	1	7	4	1
Non-causal	0	8	3	1	5	3

The bar graphs of the compression rates for the subbands are presented in Figure 5.1 (details are listed in Table A.7, Appendix A), where the subband sizes for each image, Barbara and U1, decrease from left to right when the image's number increases, as shown in Table 4.1.



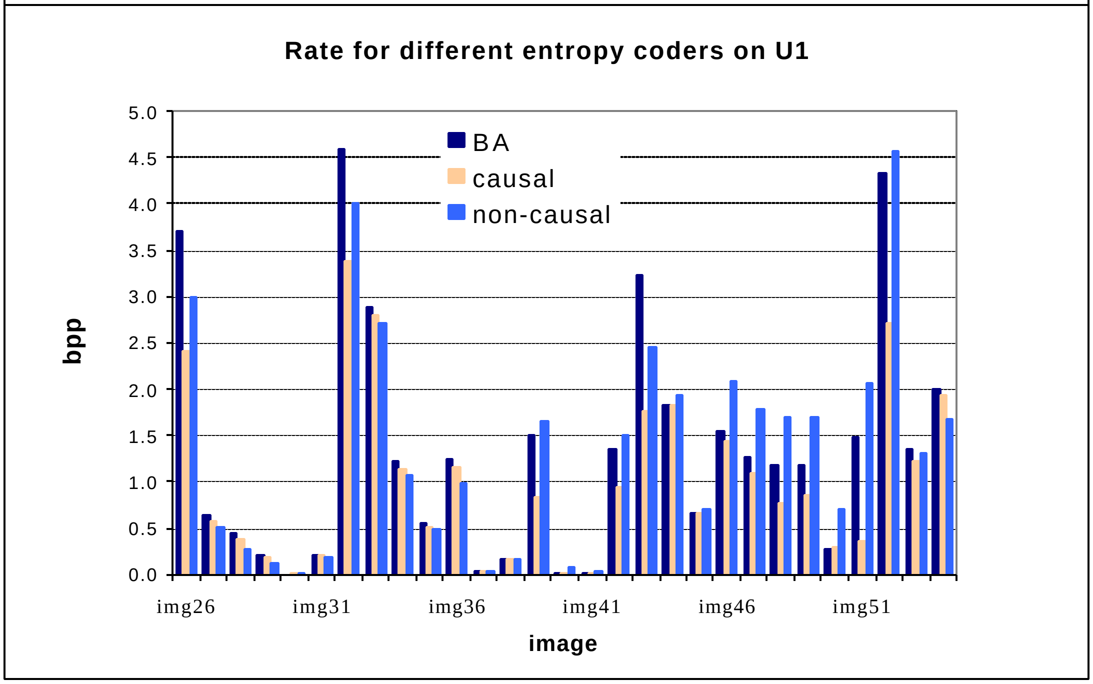


Figure 5.1 – Compression performance for different entropy coders

The basic arithmetic coder (BA) is more suitable for “small size” subbands in Barbara image. The reason for this phenomenon is that when the subband size is very small, the coder has a difficult time learning the source statistics when there is even a single extra context. Since the basic arithmetic coder cannot be treated as a subset of the causal context coder when the context size is one due to the header file issue we discussed above, it is reasonable to have the result that it is better than the causal coder on “small size” subbands. However, for the U1 image, the causal context coder outperforms the other coders on small-sized bands, which means that the appropriate context size for those subbands is not one. As a tradeoff of the compression ratio and computational complexity, we still apply basic arithmetic coder for small-sized subbands in the following experiments, however, the non-causal coder works better on large and medium size subbands for both images. These two context coders, the causal and the non-causal models, have their own advantages, and it is hard to tell which one is better simply from this experiment. In the next set of experiments, when both of them are combined with the space-frequency segmentation, it is found that the causal coder is generally superior. Recalling Section 3.4, both the basic arithmetic coder and stack-run coder are used to entropy coding the quantized SFS subband coefficients. Stack-run coding is applied in high pass subbands and the basic arithmetic coder is used for LL subbands and space only subbands. This implies that a lot of SFS subbands are coded by the stack-run coder instead of the proposed entropy coders when we try to combine them to improve the SFS algorithm. Generally, the percentage of high frequency subbands of all the subbands of a five-decomposition level SFS partition is approximately 56.4%. Later experiments in Section 5.2.2 show that substituting stack-run coder with context-based entropy coders in all the SFS subband is not a good choice, therefore we keep stack-run coder for high pass subbands.

In further experiments, context-based entropy coding is combined with the space-frequency segmentation technique, generating the modified SFS methods: the “causal_sfs” and the “non_causal_sfs” algorithms, which represent two algorithms when the causal and the non_causal context coders are applied on the SFS subbands. The basic arithmetic coder will be used on small-sized subbands, as explained above. The stack-run coder is applied on high pass subbands as in the original SFS algorithm. Details are listed in Table 5.3, where the context sizes are also included.

Table 5.3 – Entropy coders applied in the proposed algorithms

<table>
<tr><th rowspan="2">Algorithm</th><th rowspan="2">Entropy coder</th><th rowspan="2">Context size</th><th colspan="2">Subband</th></tr>
<tr><th>Natural</th><th>Ultrasound</th></tr>
<tr><td rowspan="3">Causal_sfs</td><td>Basic arithmetic coder</td><td>1</td><td>Sizes smaller than 16x16</td><td>Sizes smaller than 20x15</td></tr>
<tr><td>Causal context coder</td><td>Ranging from 1 to 10</td><td colspan="2">LL and space-only subband</td></tr>
<tr><td>Stack-run coder</td><td>N/A</td><td colspan="2">High-pass subband</td></tr>
<tr><td rowspan="3">Non_causal_sfs</td><td>Basic arithmetic coder</td><td>1</td><td>Sizes smaller than 32x32</td><td>Sizes smaller than 40x30</td></tr>
<tr><td>Non_causal context coder</td><td>66</td><td colspan="2">LL and space-only subband</td></tr>
<tr><td>Stack-run coder</td><td>N/A</td><td colspan="2">High-pass subband</td></tr>
</table>

5.2.2 PSNR Improvements with SFS Partitioning

In these experiments, we use both the causal and non-causal context models in the conditional entropy coding for SFS subbands, as shown in Table 5.3. Experiments are first implemented on general ultrasound and on ultrasound-only images, then natural images are used to obtain a general view of the compression performance. The initialization of the arithmetic coding histogram is uniform with one count for each symbol in the experiment. The number of contexts for each coder can be found in Table 5.3. The context sizes will be further modified to a specific number against different sized subbands to save execution time later in this section.

Ultrasound images

The target bit rates in the experiments range from 0.3bpp to 1.0bpp in 0.1bpp increments, and the quantizer sets change for each target bit rate[1]. Depending on the image type, the default edge value for context is 32 in both the “causal” and “non-causal” algorithms, as explained in Section 4.1.3.

Table 5.4 and Figure 5.2 illustrate the experimental results of PSNR for three different algorithms: the “SFS”, the “causal_sfs”, and the “non-causal_sfs” coders on U1 and U7. From these results, we can see that applying a context-based entropy coder on the SFS segmentation subbands achieves better compression performance than the original SFS

---

[1] These quantizer sets are from Chiu’s BSF algorithm. They are listed in Appendix C of this thesis.

algorithm using the basic arithmetic coder. The "causal_sfs" algorithm outperforms the "non-causal_sfs" scheme at the all target bit rates.

Table 5.4 – PSNR(dB) comparison on U1 and U7 (640x480)

Image	Rate(bpp)	SFS	Non-causal_bsf	Causal_bsf
U1	0.3	29.5129	29.5350	31.2238
	0.4	31.3139	31.3139	33.2037
	0.5	33.0602	33.1017	34.9246
	0.6	34.5764	34.5950	36.3847
	0.7	36.0104	36.0478	37.7603
	0.8	37.1634	37.2254	38.9507
	0.9	38.4371	38.5064	40.2335
	1.0	39.6119	39.6951	41.4390
	average	**34.9608**	**35.0024**	**36.7650**
U7	0.3	34.3917	34.3891	36.3614
	0.4	36.7668	36.7498	38.3951
	0.5	38.6747	38.7820	40.2950
	0.6	40.3642	40.4871	41.7863
	0.7	42.0228	42.2355	43.3120
	0.8	43.2800	43.5249	44.6983
	0.9	44.4869	44.8111	45.7932
	1.0	45.5833	45.7979	46.9586
	average	**40.6963**	**40.8472**	**42.2000**

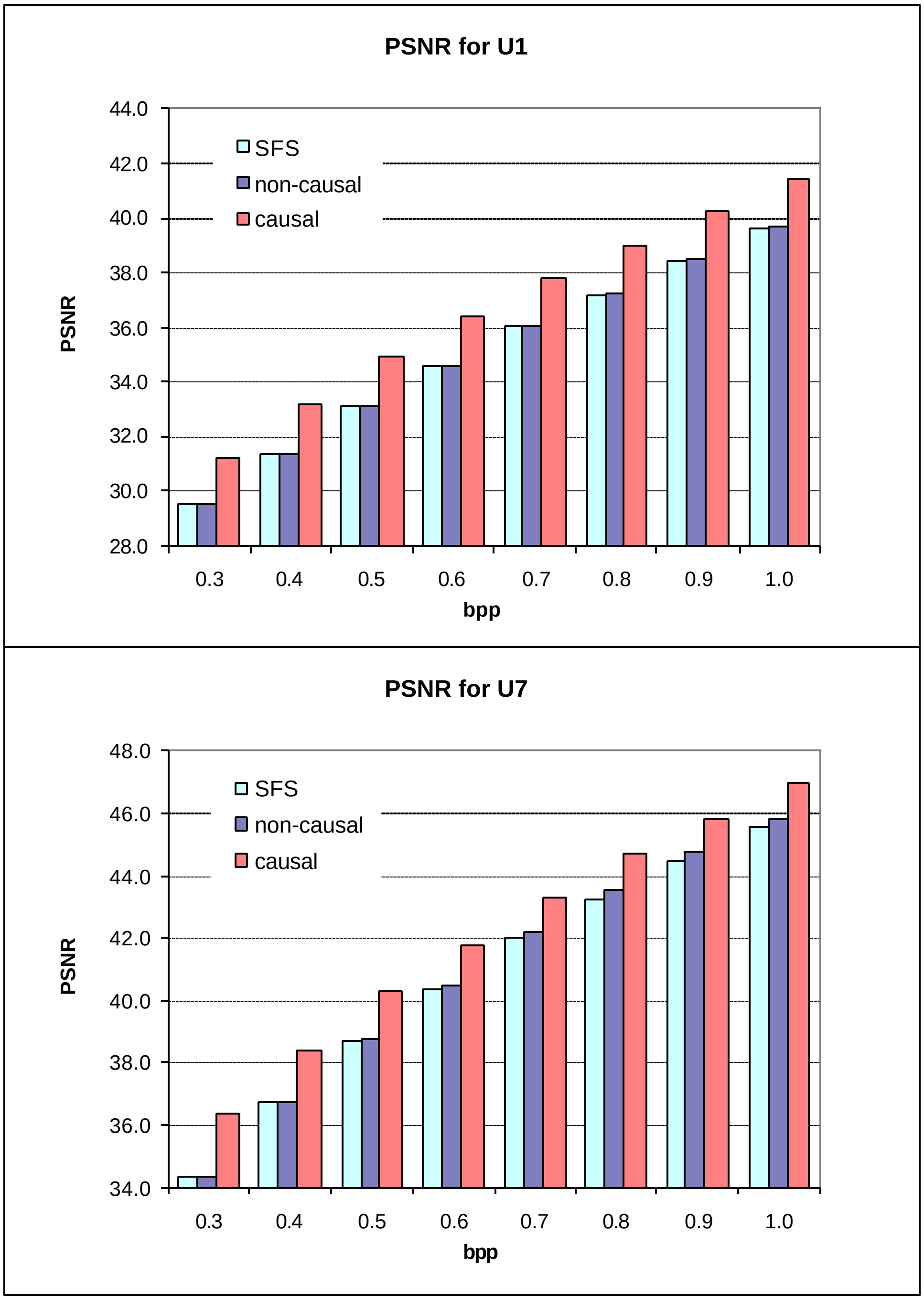
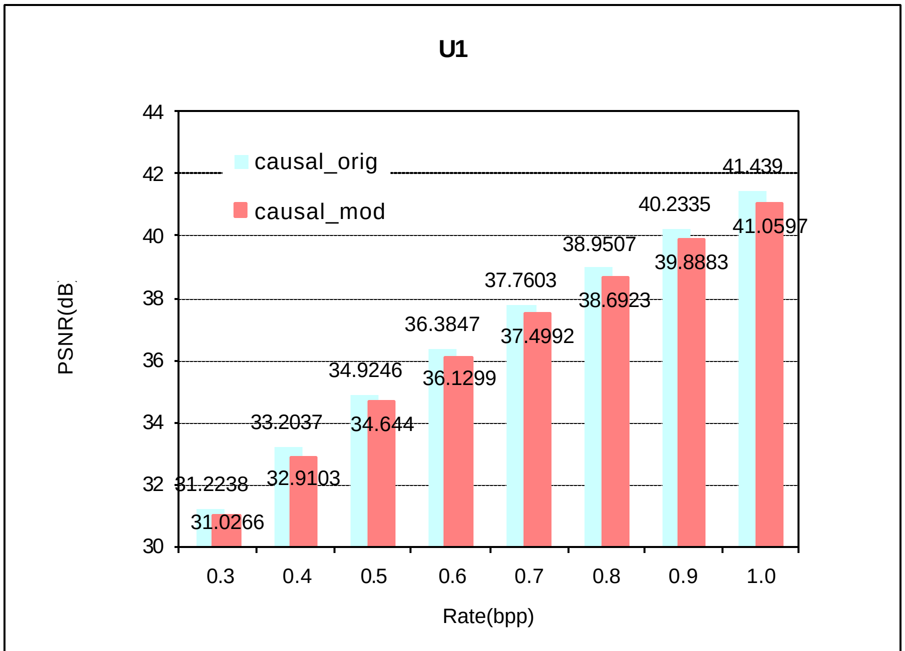


Figure 5.2 – PSNR on U1 and U7: SFS, non-causal and causal

Considering the computational complexity, we modify the causal context coder to have a fixed context size instead of optimizing the number of contexts during encoding. Based on a training set of subbands for U7 and Barbara, we obtain the optimal context size for each subband, then average those context sizes for different sized subbands. Instead of applying the context sizes presented in Table 5.3, we modify the context-based entropy coders using fixed context sizes for different sized subbands on natural images and ultrasound images. The fixed context sizes are summarized in Table 5.5. The number of contexts used for each block-size is set before the SFS segmentation is done. The compression results of the two "causal" coders are presented in Figure 5.3, where we use the term "causal_ori" to represent the original causal coder with context sizes in Table 5.3, and "causal_mod" to mean the modified causal algorithm with fixed context sizes described in Table 5.5.

Table 5.5 – Fixed context sizes for the entropy coders

Entropy coder			**Context size**
Basic arithmetic coder			1
Causal coder	Natural image (512x512)	16x16 subbands	1
		32x32 subbands	2
		64x64 subbands	7
		128x128 subbands	10
	Ultrasound image (640x480)	20x15 subbands	7
		40x30 subbands	4
		80x60 subbands	6
		160x120 subbands	10
Non-causal coder	Layered zero coding		66



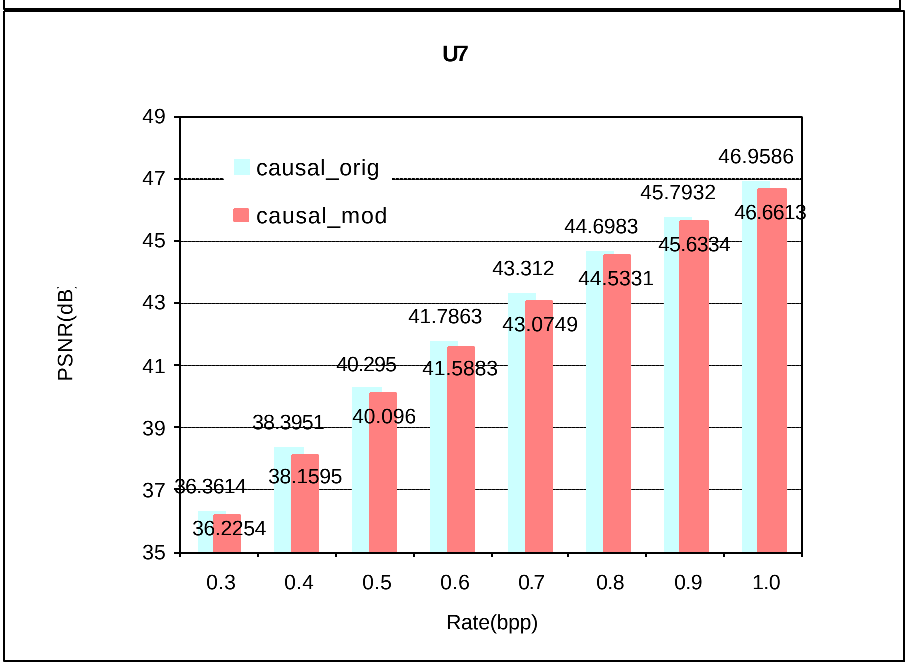


Figure 5.3 – PSNR comparison between “causal_ori” and “causal_mod”

From Figure 5.3, the PSNR values achieved from these two coders do not have much difference. The average loss of the PSNR for the “causal_mod” algorithm is approximately 0.24dB on U1 and U7, while it runs nearly ten times faster than the "causal_ori" scheme. Therefore, as a tradeoff between the compression ratio and the computation complexity, the modified causal coder is chosen as our proposed context-based entropy coder on the SFS subband to improve the space-frequency segmentation.

As mentioned in Section 5.2.1, over half of the SFS subbands are high frequency and coded by the stack-run coder instead of the proposed entropy coders in the above experiments. In order to study the performance of the entropy coders (the modified causal coder and the non-causal coder), the stack-run coder is removed from the entropy coding part in the next set of experiments. Only context-based coders are used on the SFS subbands during rate-distortion optimization, labeled as "causal_ns".

Working on U1, we apply different context-based entropy coders on the SFS subbands and have the following PSNR results shown in Table 5.6 and Figure 5.4, where "causal" means the modified causal context coder with fixed context sizes. The difference between "causal" and "causal_ns" is that "causal" coder uses stack-run coding technique for high pass subbands while "causal_ns" coder does not. Similar logic applies to the "non-causal" and "non_causal_ns" schemes.

From these PSNR results for the four different coders, we found that the "non_causal_ns" coder has similar performance with the "non_causal" coder, but it still can not compete with any of the causal context coders. The reason for the better performance on large size subbands of the non_causal coder shown in Table 5.2 is that those large size subbands are obtained at decomposition level three, instead of five, which is used in our experiments. Therefore, when the non_causal coder is combined with the SFS partition technique at decomposition level of five, most of the subbands are small and the PSNR results are not as good as we expected.

Table 5.6 – PSNR(dB) comparison on U1

Rate(bpp)	causal	Causal_ns	Non_causal	Non_causal_ns
0.3	31.0266	30.8871	29.5350	29.4740
0.4	32.9103	32.8101	31.3139	31.2421
0.5	34.6440	34.5714	33.1017	33.1422
0.6	36.1299	35.9578	34.5950	34.6347
0.7	37.4992	37.3531	36.0478	36.0526
0.8	38.6923	38.5921	37.2254	37.2735
0.9	39.8883	39.7827	38.5064	38.5511
1.0	41.0597	40.9845	39.6951	39.7301
average	**36.4813**	**36.3674**	**35.0025**	**35.0125**

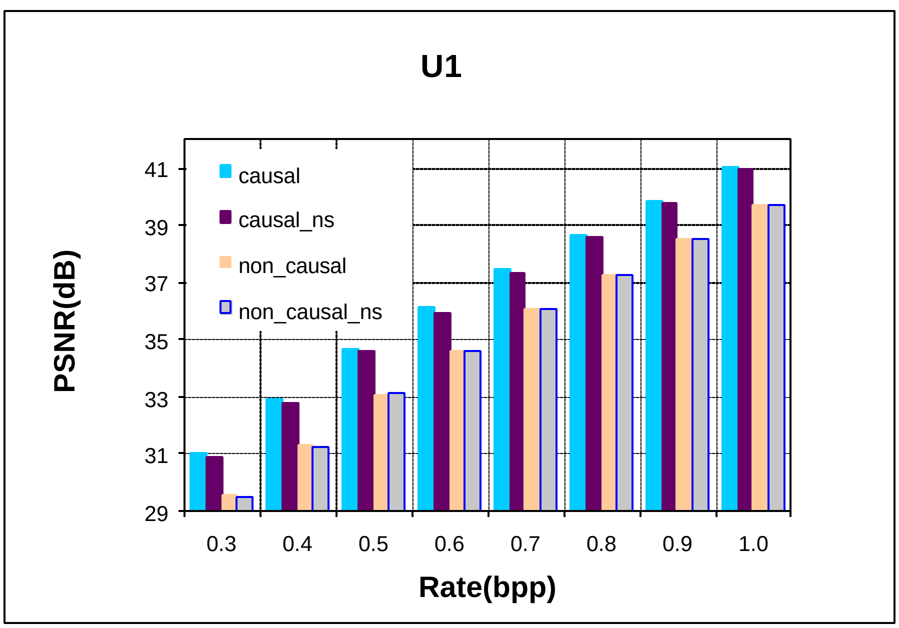


Figure 5.4 – PSNR comparison with and without stack-run coder

When the stack-run coder is removed from the entropy coding, the average PSNR for the “causal_ns” coder is 0.12dB less than that of the “causal” scheme which has stack-run coding on high pass bands. Therefore, a stack-run coder is more appropriate for the high frequency subbands than the causal context-based entropy coder. Finally, the modified causal coder with fixed context sizes for each subband is chosen as our Context-Based Entropy Coder(CBEC) on LL and space only bands to incorporate with the SFS method.

In order to see the differences between the space-frequency segmentations using different entropy coders, Figure 5.5(a) and (b) present two different SFS partitions using the original “SFS” scheme and the CBEC method described above. It is interesting to find that almost all of the small space only bands have been eliminated in “CBEC” representation, but the performance is still better, as shown in Table 5.4. We summarize the number of each type of subband in Table 5.7, where “S”, “M” and ”L” represent small, median and large size subband respectively. The definition of each size group of subband is listed in Table 5.1. This evidence implies that the context-based entropy coder works better when all the space only bands join together than when they are separated. Thus we could combine all space only subbands to form a larger band and entropy code this band with our proposed context coder, to save a fair bit of computational complexity. This work needs further scientific analysis and experimental results; we leave it as an open question for future work.

(a) SFS (b) CBEC

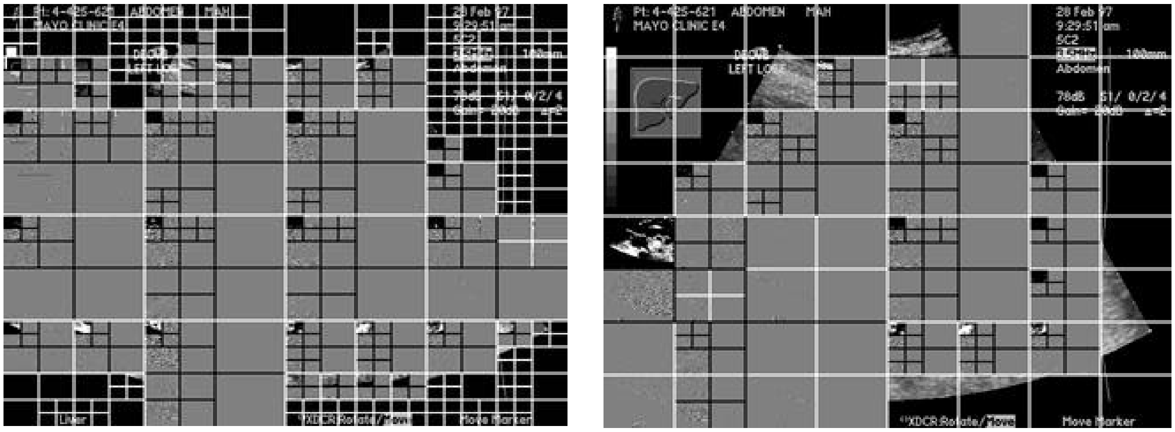


Figure 5.5 – Subband images for different coding schemes

Table 5.7 – Number of each types of subbands for SFS and CBEC

algorithm	**LL**			**High pass**			**Space only**		
	S	**M**	**L**	**S**	**M**	**L**	**S**	**M**	**L**
SFS	0	0	0	164	82	0	136	30	0
CBEC	0	0	0	84	51	0	0	40	0

Figure 5.6 illustrates the results when the proposed algorithm (the "CBEC" coder) is compared with the SPHIT and the SFS algorithms. The proposed algorithm outperforms both the SPIHT and the SFS at various target bit rates on the experimented images, U1 and U7.

The average PSNR for these coders are listed in Table 5.8.

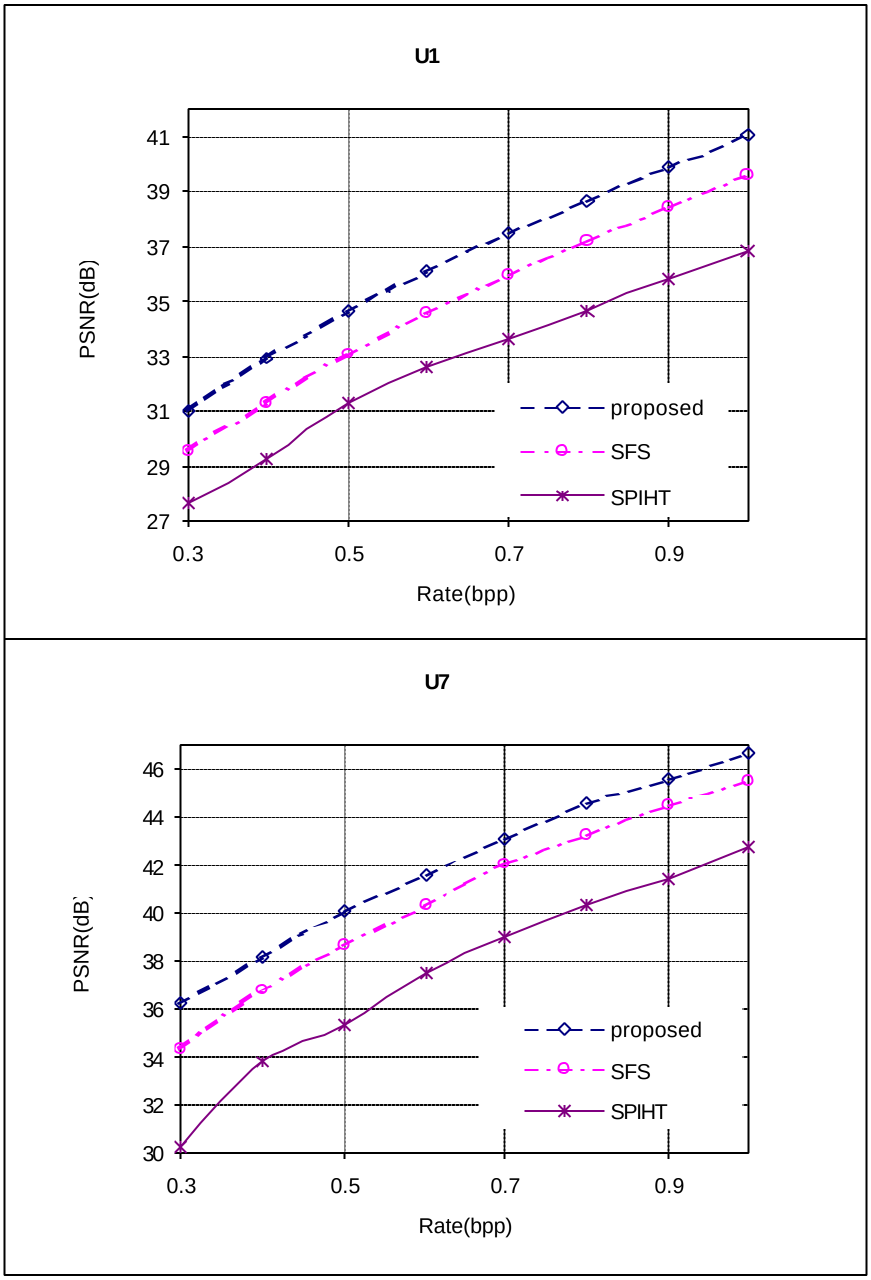


Figure 5.6 - PSNR for U1 and U7: the proposed, SFS and SPIHT

Table 5.8 – Average PSNR(dB) between 0.3pp and 1.0bpp on U1 and U7

image	SPIHT	SFS	proposed
U1	32.7188	34.9608	36.4813
U7	37.5749	40.6963	41.9965

SFS performance on Image t11 and t12

Images t11 and t12[2] are two ultrasound images which have not been included in the training set to estimate the off-line parameters (the default value for edge contexts and the context sizes for different subbands) for context modeling. When the proposed algorithm, the "causal_mod" coder with fixed context sizes, is applied on these images, the PSNR results show the advantage of the proposed algorithm over SPIHT and SFS. Figure 5.7 gives a comparison between SPIHT, SFS and the proposed algorithm on t11 and t12. It is obvious that the PSNR obtained from the proposed algorithm is better than the other two schemes at various target bit rates ranging from 0.3bpp to 1.0bpp. Numerical results of PSNR are shown in Table 5.9. The perceptual quality of these algorithms on t11 are also shown in Figure 5.8.

Table 5.9 – Average PSNR(dB) between 0.3pp and 1.0bpp on t11 and t12

image	SPIHT	SFS	proposed
t11	38.1100	41.0391	41.7335
t12	39.4205	42.3721	43.0214

The perceptual quality difference of the ultrasound image between the algorithms is hard to compare from Figure 5.8. They all look the same, but at least the optimized partition in CBEC method does not give extra artifacts for the fine details in image t11.

[2] Image t11 and t12 are obtained from the Radiology Department of Vancouver General Hospital. The original images are shown in Appendix D.

Detailed comparisons between the coders for ultrasound-only images are presented in the following section.

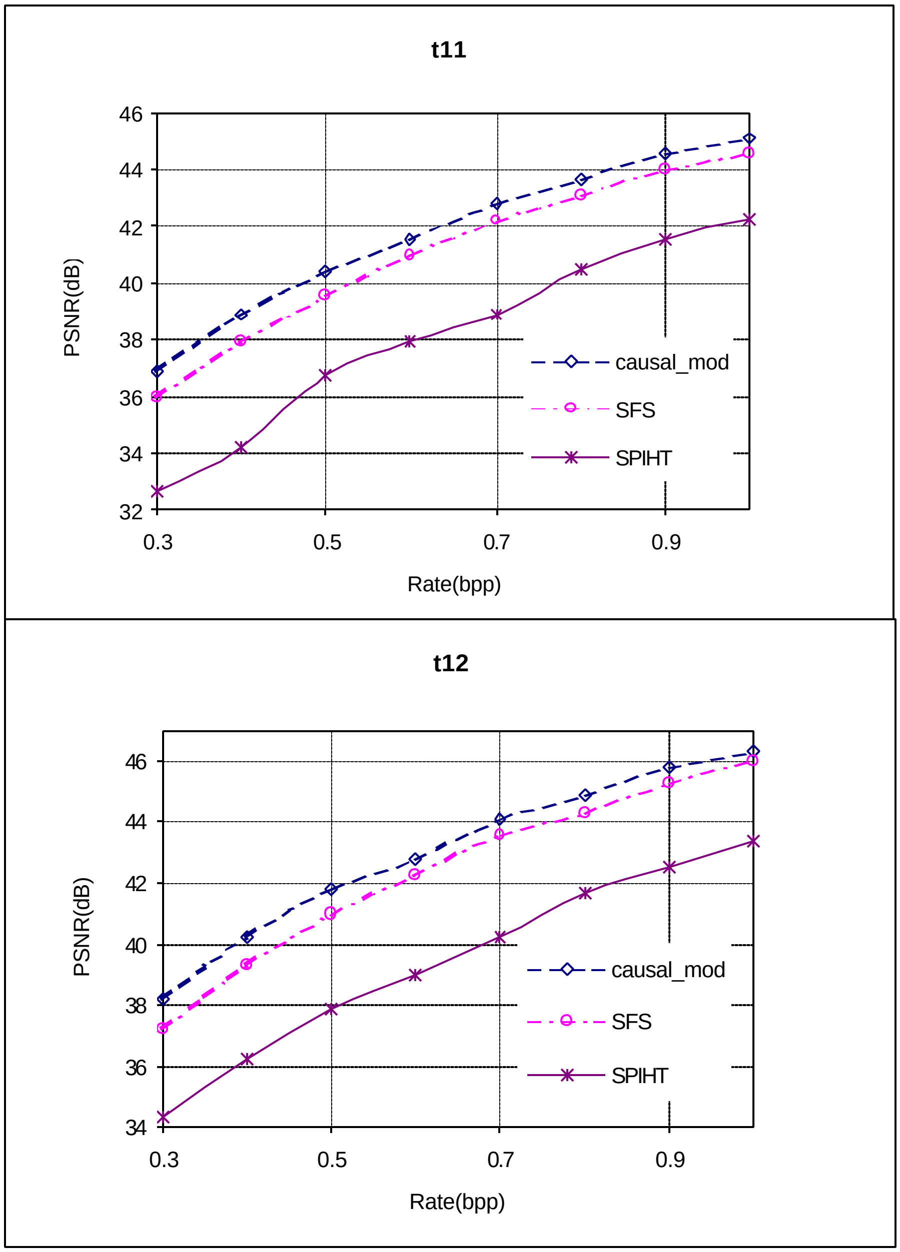


Figure 5.7 - PSNR on t11 and t12: causal_mod, SFS and SPIHT

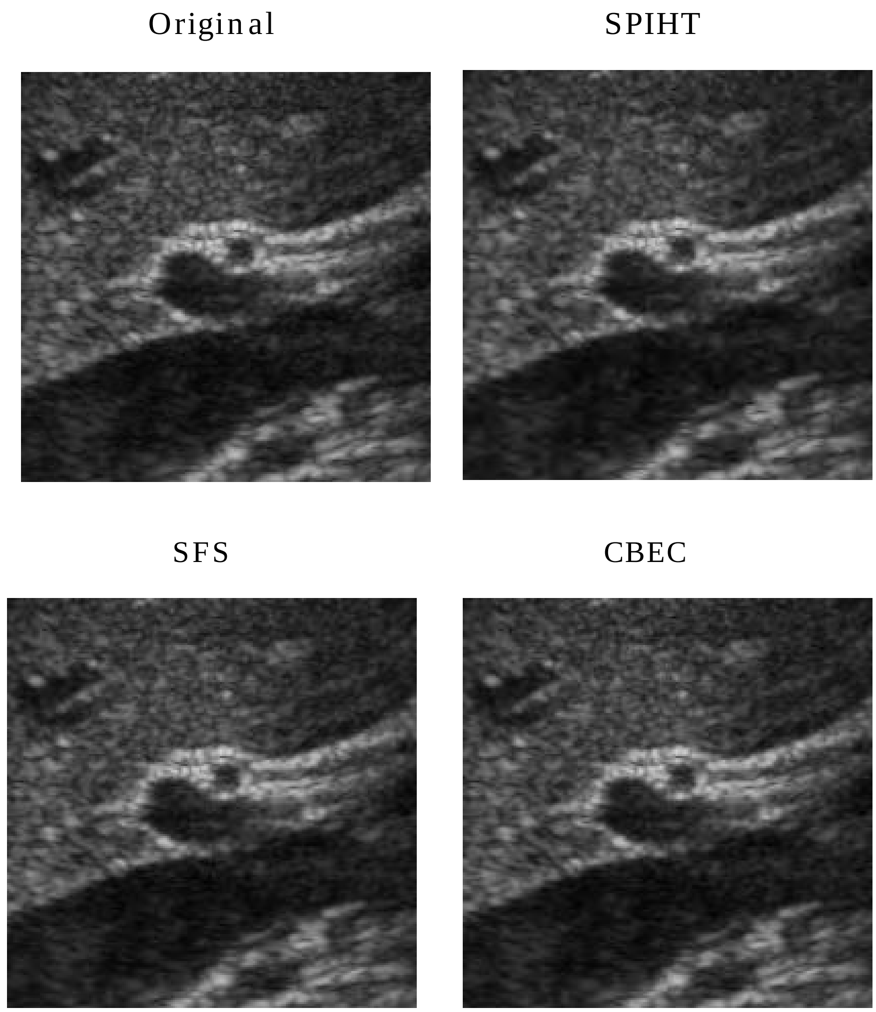


Figure 5.8 – Fine details in t11: original, SPIHT, SFS and CBEC, 0.5bpp

Ultrasound-only image (C1 to C5)

The medical ultrasound images usually have non-rectangular shapes of scanning areas and they also have textures for examination information, displaying at the corner of the images. In order to research the compression performance on the ultrasound-only images, the proposed algorithm (the causal context-based entropy coder), SPIHT, and the SFS schemes are applied on five rectangular ultrasound-only images, C1 to C5[3], which are cropped from five medical ultrasound images with wedge shaped ultrasound scanned areas.

We implement the experiments using the same ultrasound-only images and decompose them at the same level of three as described in Chiu's research [22], due to the consideration of fair comparison. The results for both the real compression rates and the PSNR values presented in Table 5.8 show that the proposed algorithm outperforms both the SPIHT and the SFS methods. Because the decomposition level is only three in this set of experiments, the partitions of the proposed algorithm and the SFS scheme are not as different as those found using the full ultrasound images in Figure 5.5. The partitions of the two algorithms for image C1 and C5 are quite similar, whose PSNR values are also similar as shown in Table 5.10. There are some differences between the partitions for images C2, C3 and C4. The CBEC algorithm eliminates the small-sized low frequency bands in C4, but tends to have more frequency/space partition in LH bands than the SFS scheme, and better PSNR results. The common partition for both algorithms is that they all have a frequency decomposition at first, and then continue to have frequency partition in LL and LH subbands. The space partitions only happen at the third decomposition level. Because the stack-run coder is used on high pass subbands, as mentioned in Section 5.21, the percentage of the bands using entropy coder, the LL band and the space-only bands, among the whole subbands is also estimated in Table 5.10.

---

[3] The five ultrasound-only images are shown in Appendix E . They have the same size of 192x256.

The partition representatives for the SFS scheme and the proposed algorithm (the CBEC) are presented in Appendix E together with the original images C1 to C5.

Table 5.10 – Compression performance vs. SPIHT & SFS

Image	Algorithm	Rate(bpp)	PSNR(dB)	Percent(%)
C1	SPIHT SFS Proposed	0.5000 0.4829 0.4766	30.9860 31.4462 31.4462	N/A 5.26 5.26
C2	SPIHT SFS Proposed	0.5000 0.4893 0.5010	31.3897 31.6178 31.9008	N/A 40.9 22.7
C3	SPIHT SFS Proposed	0.5000 0.4697 0.4950	32.6403 32.6351 33.1373	N/A 5.26 4.55
C4	SPIHT SFS proposed	0.5000 0.4992 0.4948	34.2514 34.4473 34.5033	N/A 26.3 47.4
C5	SPIHT SFS proposed	0.5000 0.5000 0.4912	33.4025 33.6784 33.6784	N/A 5.26 5.26

Natural image (Airplane and Bridge)

Similar to the experiments for ultrasound images, we now compare the PSNR values among SPIHT, SFS, and the proposed algorithms (the CBEC coder) on natural images, Airplane and Bridge(512x512), which are shown in Appendix D. The CBEC coder has been discussed above and the context sizes for the causal coder is fixed to a certain number as listed in Table 5.5. Table 5.11 presents the PSNR values using these three algorithms. A clearer view of the difference between the proposed algorithm with the SPIHT and the SFS can be obtained from Figure 5.9.

Table 5.11 – PSNR(dB) on Airplane and Bridge, 512x512

image	Rate(bpp)	SPIHT	SFS	proposed
Airplane	0.20	31.1764	31.1777	31.2399
	0.25	32.4633	32.3152	32.3556
	0.50	.36.5364	36.1185	36.1743
	1.00	41.1584	40.1750	40.4240
	average	**35.3336**	**34.9466**	**35.0485**
Bridge	0.20	24.3317	24.1105	24.0491
	0.25	24.8618	24.7018	24.7244
	0.50	27.0407	26.9119	26.9694
	1.00	30.4396	30.2231	30.3066
	average	**26.6685**	**26.4868**	**26.5124**

The performances for the proposed algorithm is better than SFS for various target bit rates between 0.2bpp to 1.0bpp as shown in Table 5.9, meaning that the context-based entropy coder really works for the SFS partition on natural images. However, it is not as good as the SPIHT algorithm, although the average PSNR for the proposed algorithm is only around 0.7% less than the SPIHT for both images. This implies that when the space-frequency segmentation is compared with the SPHIT method, it has not any advantage as it does for ultrasound images.

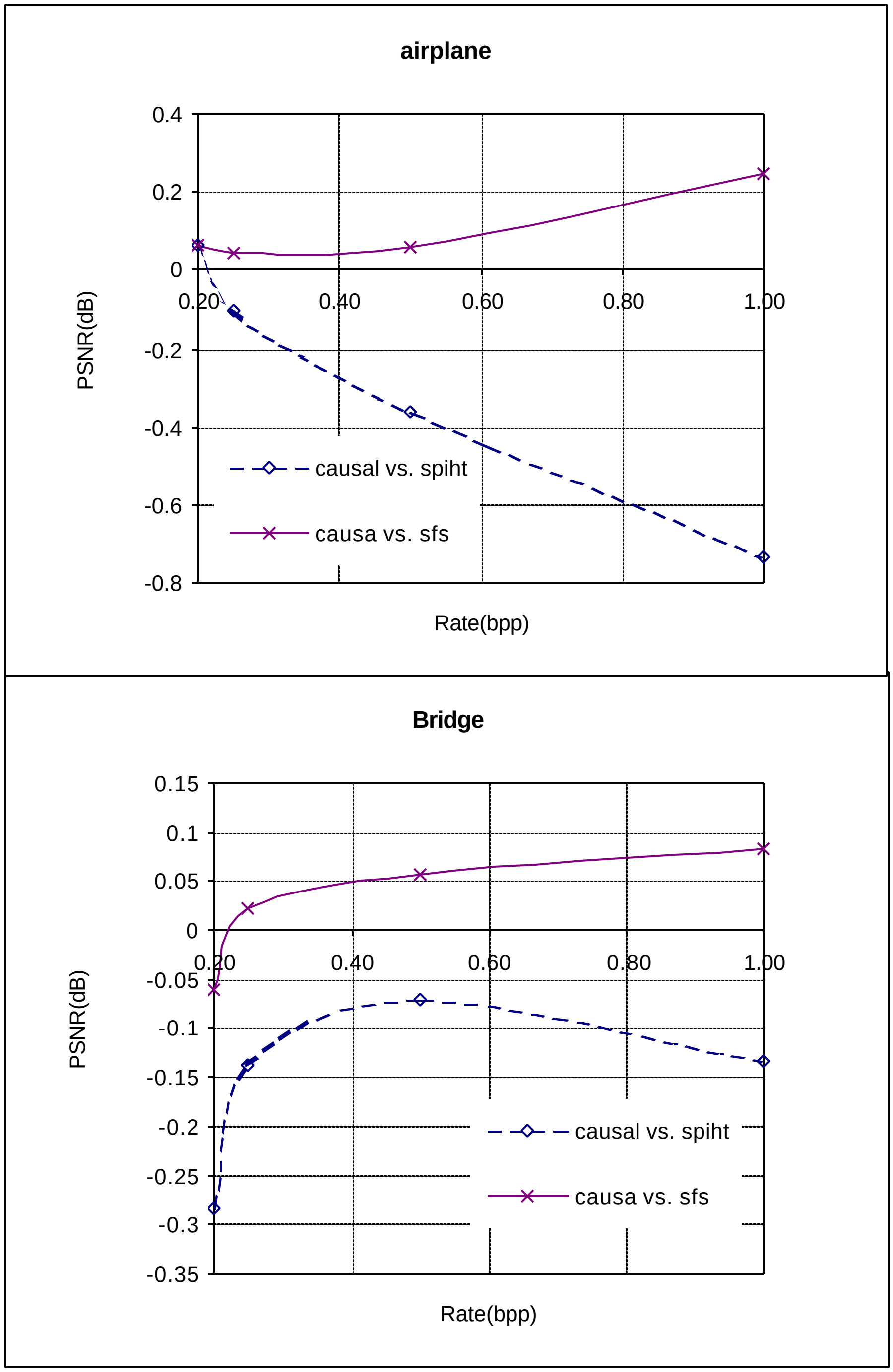


Figure 5.9 – PSNR comparison on Airplane and Bridge vs. SFS & SPIHT

Chapter 6

Histogram Initialization

Arithmetic coders typically use cumulative frequency histograms (CFH) to estimate the probability distribution of the symbols. The initialization of the histograms, which affect their learning quality, leads to different compression performance of the coder. In this chapter, we analyze the statistical distribution of the subband coefficients in order to apply modified histogram initialization suiting the practical probabilities. In the adaptive arithmetic coder applied here, the initial count for each symbol in the CFH is zero. When a symbol k is first coded by this arithmetic coder, its distribution cannot be estimated using this histogram since its frequency is zero. Therefore, we initially assumed that it is uniformly distributed with the initial count of one, and the initial probability p_{ini} of each symbol equals one over the number of symbols in the source data:

$$p_{ini} = \frac{1}{nSymbol} \qquad 6.1$$

Then, as the symbol k occurs more often in the input stream, its probability distribution $p(k)$ is estimated to be:

$$p(k) = \frac{f_c(k)}{\sum_i f_c(i)} \qquad 6.2$$

where $f_c(i)$ is the running count of symbol i. Because the statistical distributions for the subbands are quite different in the real world, uniform initialization is applied to keep an conservative way for the efficiency of the arithmetic coder in the original SFS algorithms. In order to improve the rate of adaptation, we used a training set of subband images and computed the associated histogram for each subband and each context. The initialization of the histogram is then intended to be adapted according to the expected statistics.

Usually, the histogram (CFH) for the adaptive arithmetic coding is initialized to be uniform. All the experiments presented in Chapter 4 and 5 have a uniformly initialized histogram. Since the real statistical distribution for each subband is not uniform, the histogram initialization procedure for each coder can be potentially improved via a non-uniform histogram initialization. However, modifying the initialization for the histograms in an arithmetic coder is a complex task, since the statistical distributions for the subbands are quite different from each other and it is impractical to modify the histograms of each subband with respect to its own statistical structure. The experiments in this chapter are designed to find a general histogram from a training set based on the analysis of the probability of the SFS subbands, which might improve the compression results.

The statistical distributions of the data in the low frequency and high frequency subbands are different. We try to pick two sample subbands among our 54 experimental subbands listed in Table 1 of Appendix A, standing for low and high frequency respectively to analyze their statistical distributions. Because all the small size subbands (16x16) from the fifth SFS decomposition level are high frequency, image16 and image 19 are randomly selected among the 32x32 sized subband group. The SFS subband coefficients are quantized to integers, from 1 to 64, as explained in Section 4.1.5. Therefore our histograms have 64 symbols in total, where the index 32 corresponds to the value of zero in unquantized subbands. The distributions are obtained from the counts of each symbol in these two subband images and depicted in Figure 6.1 where the “x” axis represents the symbol; and the “y” axis means the counts of occurrence for each symbol in the subband. In the meantime, general histogram distributions for the low and high frequency subbands at different subband sizes are constructed upon the entire experimental SFS subbands described in Section 4.1.5. The initial count, c_k , for the symbol k in the general histogram for the low frequency subband is estimated by a summary of the counts in all of the low frequency subbands of the same subband size, bin by bin, and then averaging the initial histogram by dividing by the number of subbands, $nSub$, being counted.

$$c_k = \frac{\sum_{j=1}^{nSub} c_{k,j}}{nSub} \qquad 6.3$$

where $c_{k,j}$ is the count for the symbol k in the j^{th} subband. A similar procedure is done for the high frequency subbands. The reason for generating the average histogram in this way is because we assume that the data structure for each subband is quite different. Under a conservative consideration, we average the counts among the subbands with the same size, trying to find their common characteristics of statistical distributions. The estimated initial counts for the general and high histograms are illustrated in Figure 6.2. The normalization of the histograms is set to one, which means the initial counts are exactly the same average value from the example subbands. We will adjust the normalization according to the compression performance of the entropy coder later in this chapter.

Because there is only one low frequency subband among the 32x32 sample subbands, the general histogram is exactly the same as that of image 16. For the high frequency subbands, the average distribution has a similar shape as image 19, although the numerical counts in the histograms are not exactly the same.

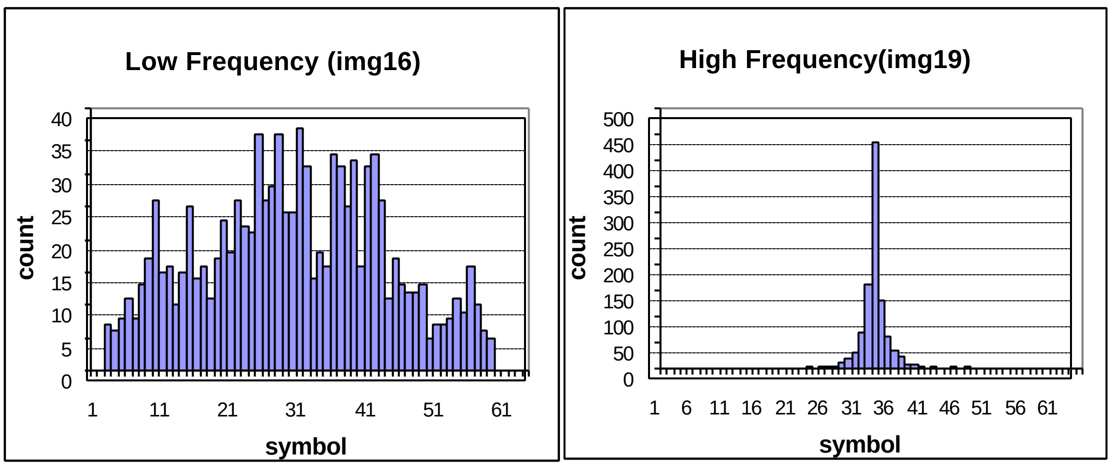


Figure 6.1 – Distributions of img 16 and 19

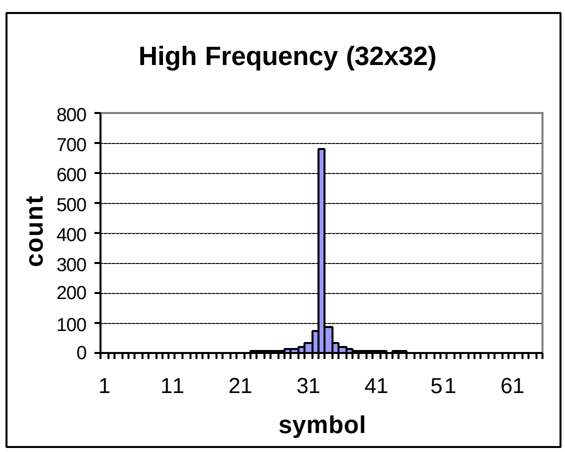


Figure 6.2 – General distribution for high pass subbands (32x32)

Comparing these two set of histograms, we found that they are quite similar in distribution. Therefore, it is reasonable to assume a practical method to apply a general histogram instead of each subband's specific distribution for the initial CFH of the adaptive arithmetic coder. The following experiments aim to demonstrate this assumption. However, before we evaluate the entropy coders with modified histograms, we try to adjust the normalization factor for the initial counts of the histograms according to their compression performance. The basic arithmetic coder with modified CFH initialization at different normalization factor is applied on the quantized SFS subbands, the images 16 and 19. Different compression rates are achieved in different cases. An ideal factor will finally be selected in this set of experiments. The relationship between the compression rate and the normalization factor is shown in Figure 6.3.

Recalling Section 2.1.1, the best an unconditional entropy coder can do is to compress the source with an average number of bits equal to the entropy of the data. So we estimate the zeroth entropy H_{img16} from equation (2.1) for image 16, the low frequency subband in our experiment, to compare with the rates obtained from the entropy coder we are evaluating.

$$H_{img16} = -\sum_{i=1}^{64} p(i) \log p(i) = 5.6506 \qquad 6.4$$

Similarly, the zeroth entropy H_{img19} for image 19 is 2.8395.

From Figure 6.3, we can see the compression rates for both the two images are higher than their zeroth entropies. For the high frequency subband, the compression rate reaches the lowest point, 2.9402, when the normalization factor is 0.5. For the low pass subband, the bit rate goes down as the normalization factor increases. We keep on testing it and finally find that when the factor is 1000, the compression rate gets to be the lowest, 5.6530bpp. Therefore, in the following experiments, we normalize the histogram initial counts by a factor of 0.5 and 1000 respectively for high and low subbands.

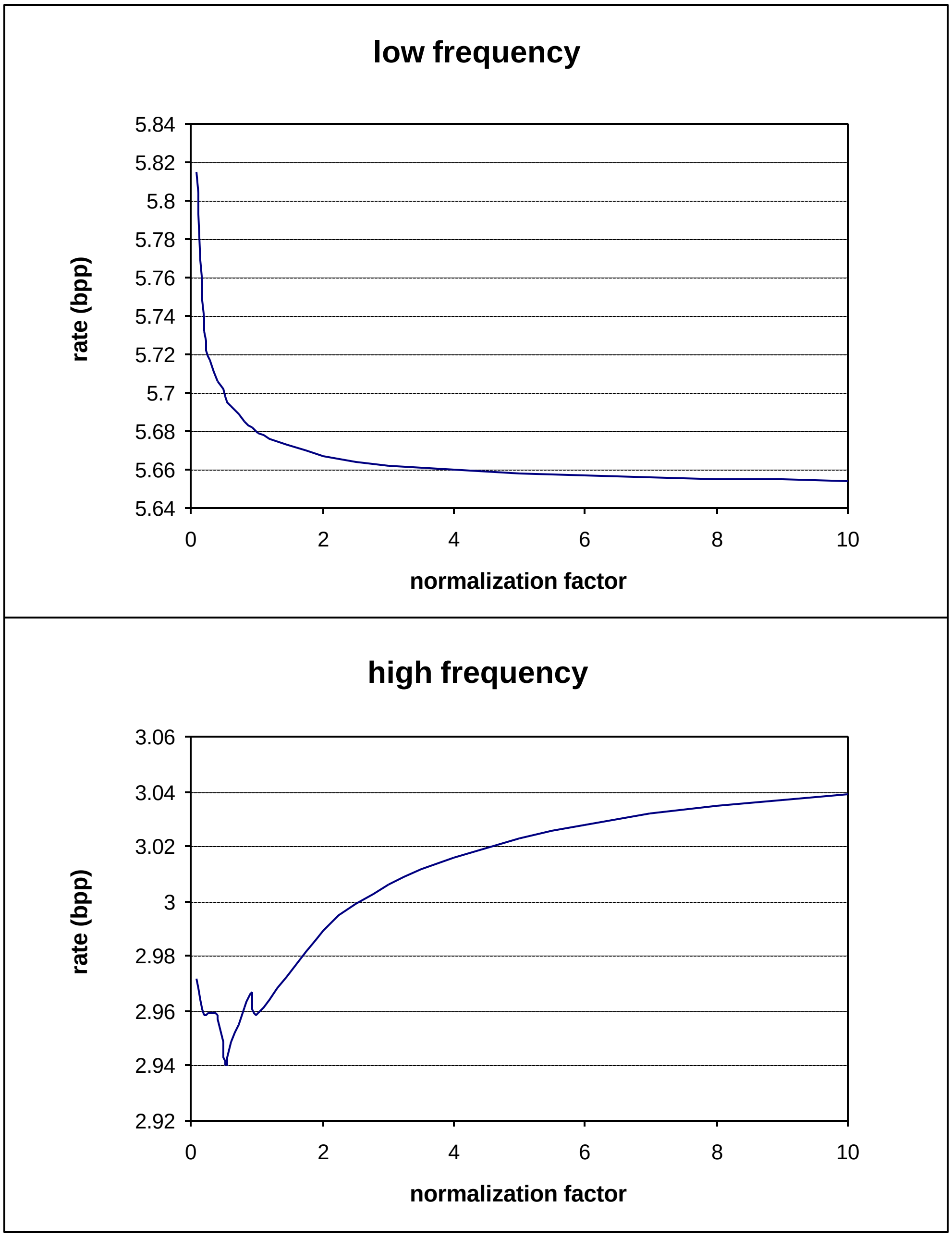


Figure 6.3 – Rates vs. normalization factors

Modifying the initial counts for CFH with the distributions shown in Figure 6.2 and normalizing them as stated above, several entropy coders are tested to see whether the histogram initialization affects the compression performance. We use the subband images 16 and 19 in this set of experiments, standing for the low and high frequency subbands. Figure 6.4 shows the compression rates of the following compared entropy coders.

BA -- The basic arithmetic coder with a uniform distributed CFH initialization, where the initial count for each symbol is one.

BA_m -- The basic arithmetic coder with modified CFH initialization shown in Figure 6.2. The initial counts are normalized by a factor of 0.5 or 1000 for the high or low frequency subbands.

CBE -- The context-based entropy coder with uniformly distributed initial CFH in the adaptive arithmetic coder. The "context" model applied here is the causal model selected from the previous experiments with a fixed number of context states as listed in Table 5.5. The initial counts in the histogram is one as stated before.

CBE_m -- The context-based entropy coder with modified initial CFH. The context model is same as the "CBE". For each context, the modified initial counts in the histogram is from the distribution of the subband shown in Figure 6.2. The normalization of the initialized histogram keeps the same as "BA_m" coder.

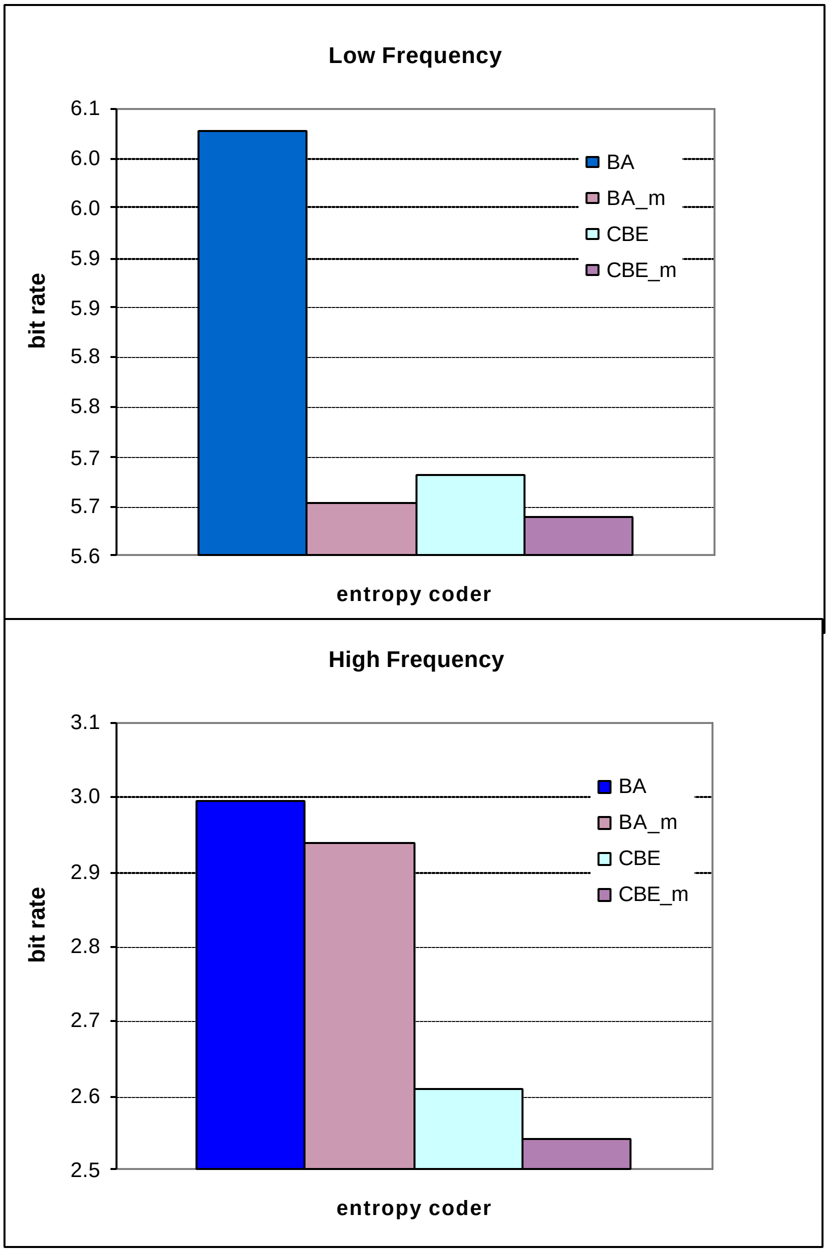


Figure 6.4 – Bit rate comparison on image16 (top) and image19 (bottom)

The rate for the "BA" coder on the image 16 is 6.0268bpp, which is higher than the entropy, 5.6506bpp calculated from (6.4). The "BA_m" coder can achieve a much better result, 5.6530bpp, indicating that the modified histogram initialization works for this sample subband. Similarly for image 19, the rate for the "BA_m" coder is 2.9402bpp, less than that of the "BA" (2.9968bpp) but larger than the entropy (2.8395bpp). For both images, the "CBE_m" coder, which considers memory among the source, achieves the lowest rate. From this set of experiments, we can see that the general histograms work for the entropy coders on the two sample subbands, especially in context-based entropy coders.

It seems promising to further study the modification of the histogram initialization at different contexts in the CBE algorithm. Therefore, in the following experiments, context-based distributions obtained from subband images 16 and 19 are shown in Figure 6.5. From Table 5.5, we know that for the 32x32 subbands, the context size is two. Thus the counts in the histograms are obtained at a context state of one or two. Like in the previous set of experiments shown in Figure 6.2, general histograms of high frequency subband at different context states are presented in Figure 6.6.

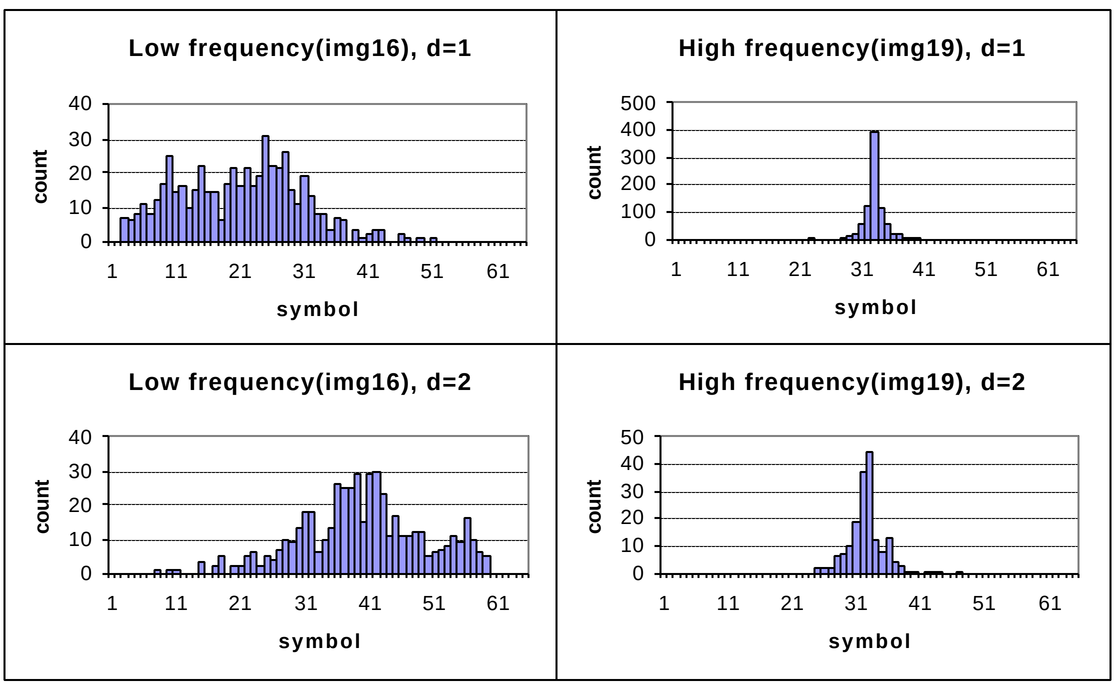


Figure 6.5 – Distributions at different contexts for img 16 and 19

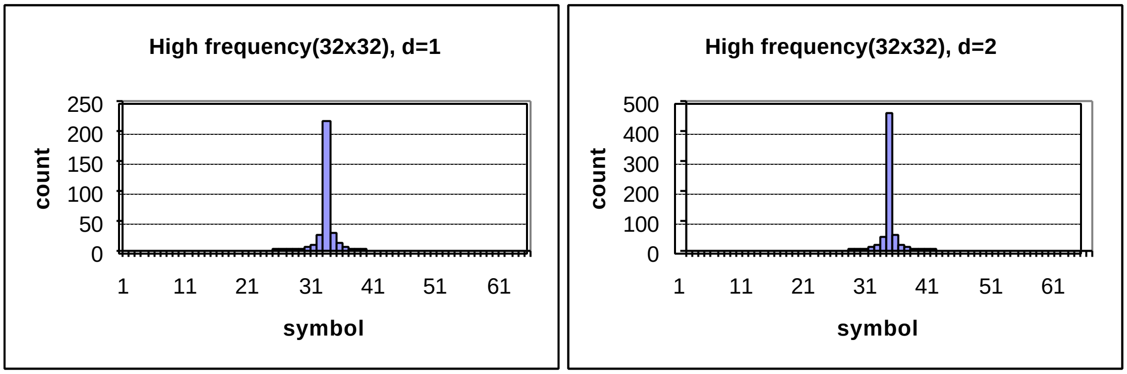


Figure 6.6 – General distributions for high pass subbands, 32x32

Working on more 32x32 sized subbands of Barbara, we have produced a general set of histograms at different context states, one or two. The method of obtaining the histograms is exactly the same as the way we obtain a general histogram without any context in the previous experiment. The set of general histograms for 32x32 sized subbands shown in Figure 6.6 have exactly the same distribution for the low subband due to the only one sample subband of 32x32. The general high pass subbands histograms are similar, but "peakier" than the distributions of the image 19 in Figure 6.5.

Normalization is necessary for the general histograms in the entropy codec. The relationship between the normalization factor and the compression rate of the context-based entropy coder with context-based histogram modification ("CBE_cm" coder) is presented in Figure 6.7, where the context model is the modified causal model with context sizes listed in Table 5.5, as described in Section 5.2.2. From the experiments, the most appropriate nomalization factor is 1500 for the low frequency band and one for the high pass bands. Finally, the normalized "CBE_cm" scheme is used on subband images 16 and 19 to compare with the other entropy coders shown in Figure 6.4. The results of the compression rates are illustrated in Figure 6.8.

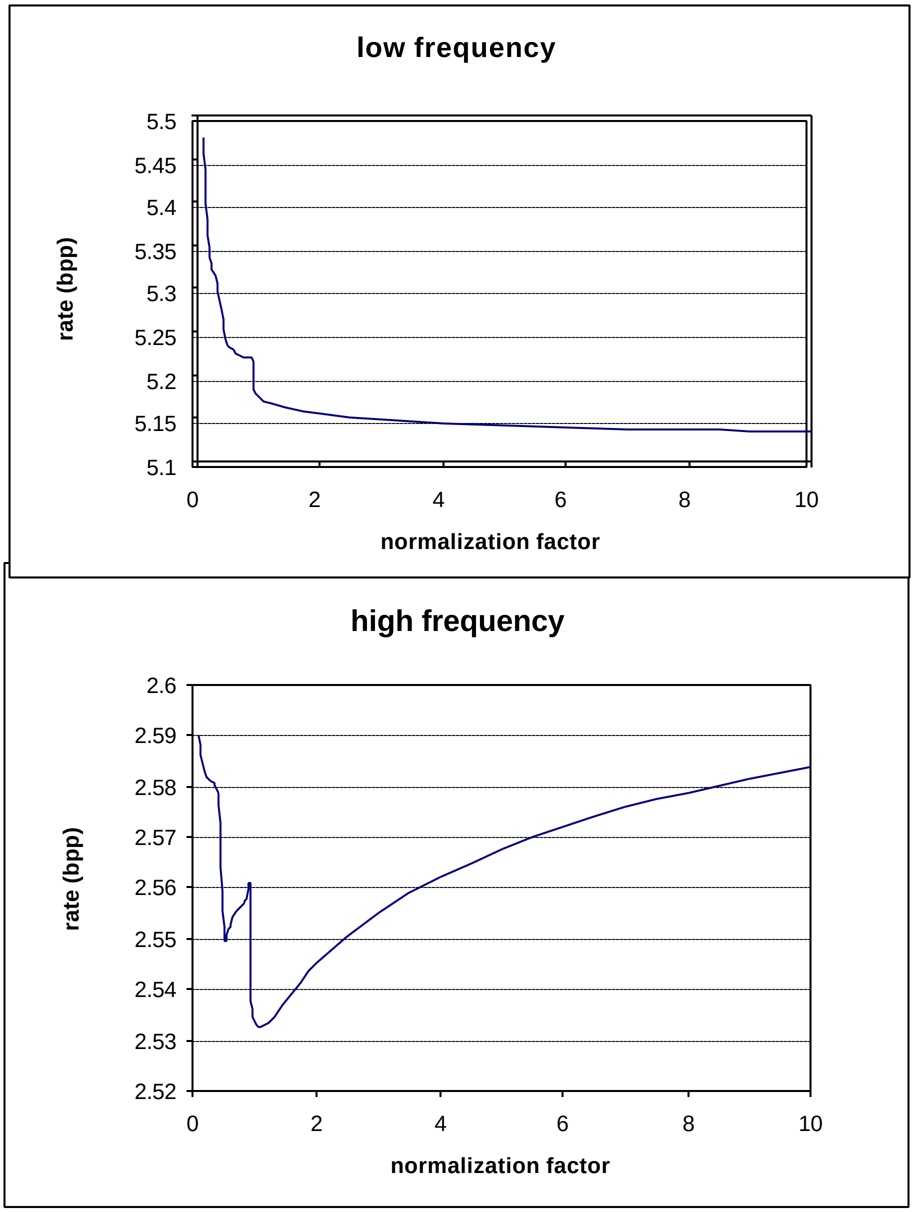


Figure 6.7 – Rates vs. normalization factors in "CBE_cm"

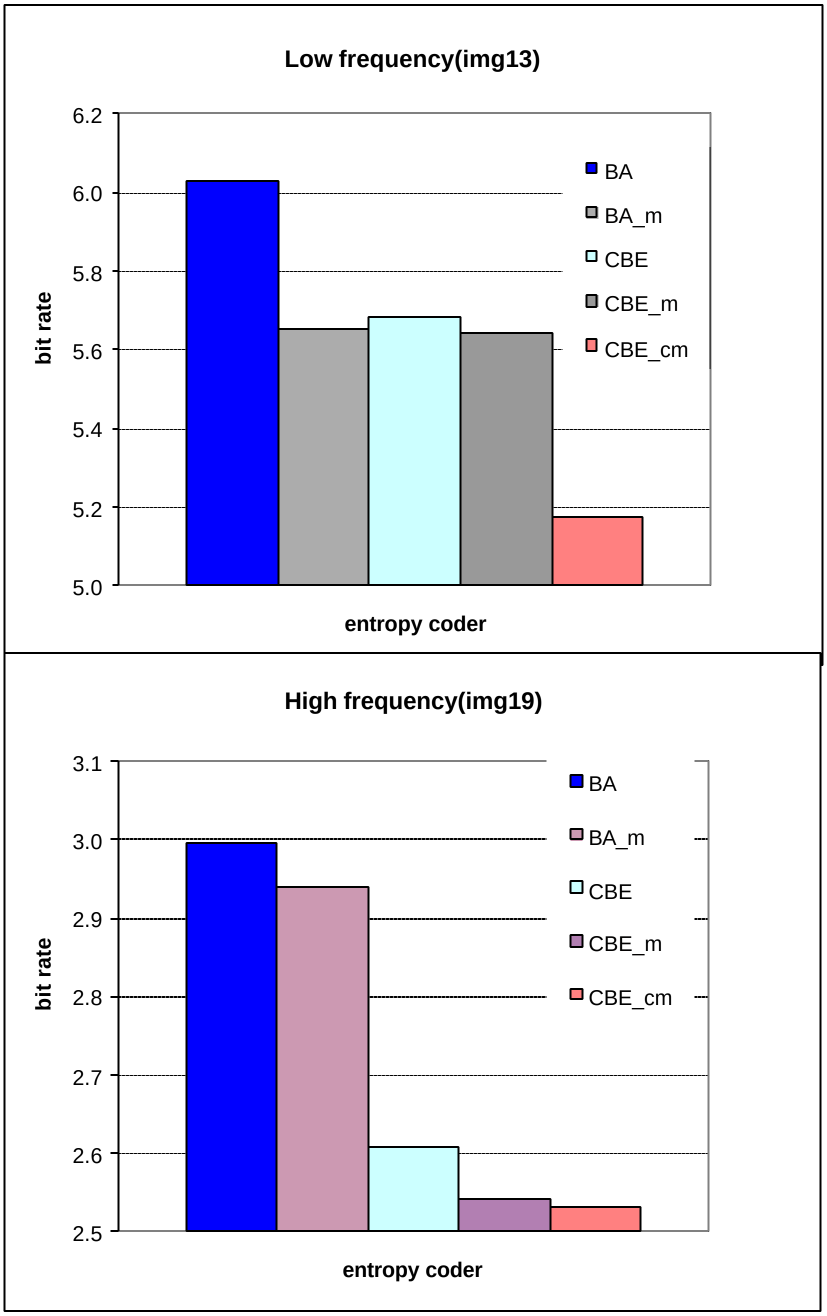


Figure 6.8 – Bit rate comparison between different coders

The “CBE_cm” coder achieves the lowest compression rate among the schemes, which means that the context-based histogram modification works for the two sample subbands. Testing on other 32x32 high pass subbands, for example img 21, we also get similar comparison results, as shown in Figure 6.9, which confirms the contribution of context-based histogram initialization to entropy coders.

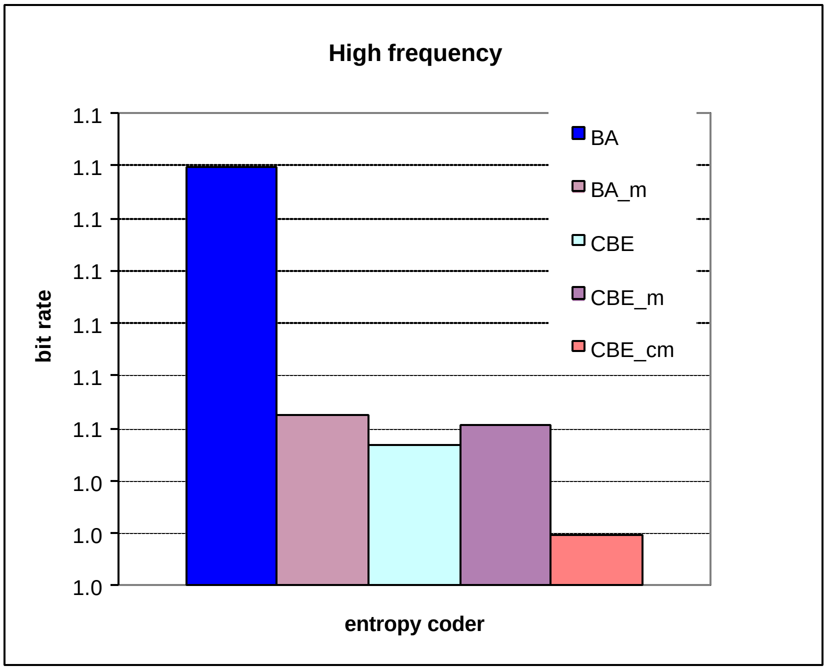


Figure 6.9 – Bit rate comparison for high frequency subband

Using the general histograms as shown in Figure 6.6 for high pass bands and Figure 6.5 for low pass bands, we modify the initial counts of the CFH for subbands larger than or equal to 32x32 in the context-based entropy coder (the "CBEC" coder described in Section 5.2.2) on the images of "Airplane" and "Bridge" to evaluate the efficiency of the context-based histogram modification. The normalization factor is the same as we used in the above experiments. The PSNR results for both images "Airplane" and "Bridge" at various target bit rates are not better than the original CBEC coder with uniformly initialized histogram, the average PSNR is the same for the two coders. There are several reasons, one is that the image "Barbara" has higher energy in high pass subands than general natural images. Thus the histograms from the training set of "Barbara" can not represent a general natural-image histogram to suit the "Airplane" and "Bridge". Other reason is that we apply the general context-based histograms shown in Figure 6.5 and 6.6 to all the subbands. It is a quite complicated issue due to the sensitivity of the probability distribution at different context states with different context sizes. We have a brief discussion about that as follows.

Context size in the histogram modifiation

The context sizes are taken into consideration for the context-based histograms in this experiments. It is intuitive that when the context size is different, the distribution will also be different although at the same context state. For example, the distribution at the third context state with the context size of ten is different from that with a context size of four. The following histograms in Figure 6.10 are generated from the same data source as before but with different context size. The term "Distribution (d=3 / 10)" means that the distibution is obtained when the context state is three with a context size of ten. We only present the distributions when the context state is three, but with different context sizes. The difference between these two sets of distributions with different context sizes shows the complexity of the histogram initialization for the entropy coder.

Overall, the experimental results in this section show that modifying the histogram initialization to improve the compression efficiency of entropy coder is a promising, but multifaceted technique for further research.

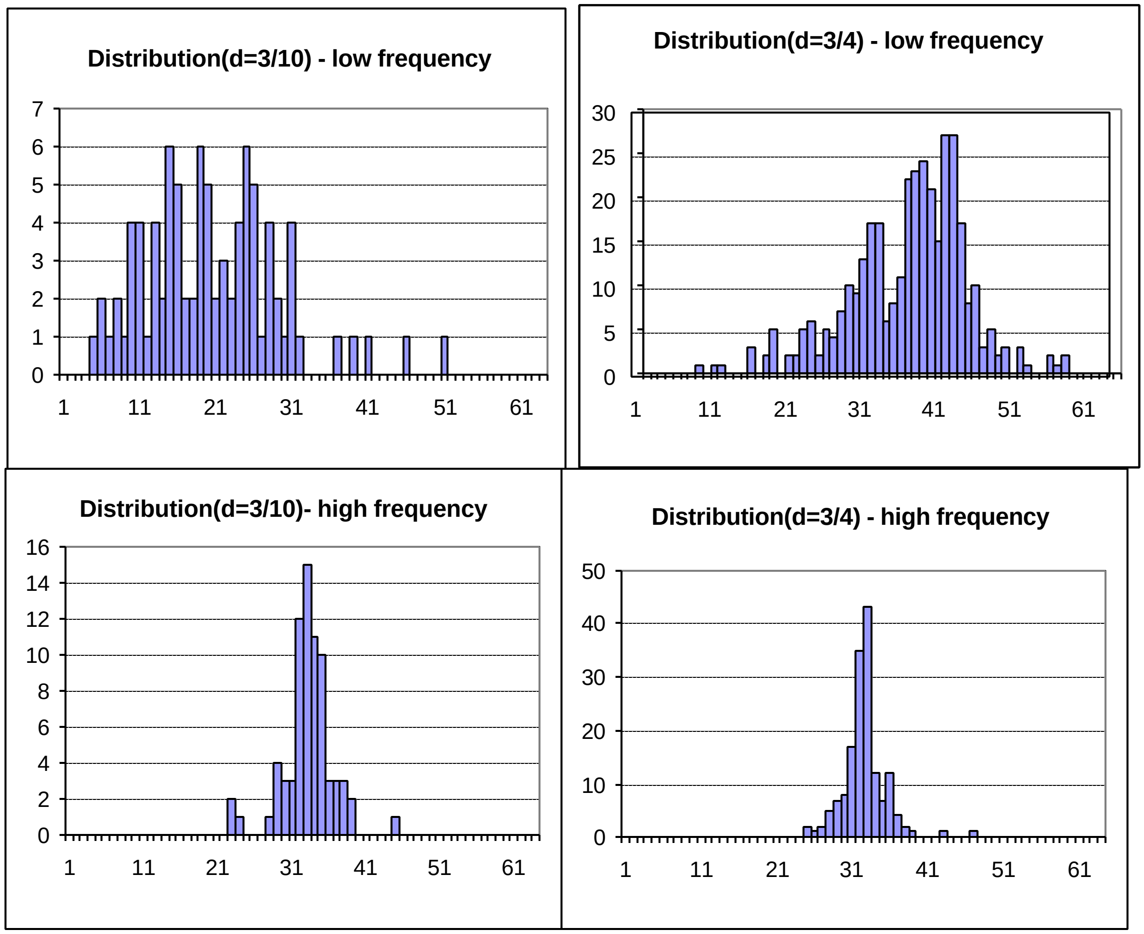


Figure 6.10 – Distributions with different context size

Chapter 7
Conclusions

In this thesis, a causal prediction based context model is developed for efficient entropy coding of images. The platform in which we used to evaluate the proposed algorithms is the compression of ultrasound images. Ultrasound image coding is different from other image coding techniques due to the unusual features of ultrasound images, such as their speckle texture and non-rectangular scan areas. The SFS decomposition method is applied in this thesis as the basis for context-based entropy coders. The causal context model with a fixed context size for different sized subband, selected as the proposed algorithm, exploits the dependencies in each SFS subband. Keeping the same quantization error, the higher-order entropy estimated from this context model achieves a higher compression ratio than the basic arithmetic coder used in the original SFS scheme. The space-frequency segmentation based on the rate-distortion performance of the entropy coder is then improved during the optimization procedure. Hence, higher PSNR values are achieved at every target bit rate when the proposed algorithm is compared with SPIHT and the original SFS schemes.

7.1 Summary of contributions

This thesis has focused on finding an efficient context-based entropy coder for the space-frequency segmentation compression on ultrasound images. The representation of the image is optimized by a rate-distortion function based on the entropy coding of each subband. The novel contributions and findings of this thesis are summarized as follows.

The context-based entropy coding techniques have been applied in the lossless coding for natural images by a number of researchers. Differential Pulse Code Modulation (DPCM) is usually used in these coders to achieve higher compression efficiency. In our proposed research, the prediction errors of each pixel are not entropy coded because the quantized SFS bands no longer have the same statistical structures as continuous-tone natural images. The values of the quantized subband is only between zero and 64 instead of 256, and 32 becomes the dominant symbols among the SFS subband indices. In the proposed algorithm, the predictor of each symbol is selected from several choices with respect to its orientation. The adaptive methodology makes the proposed context model more powerful than other popular entropy coders discussed in this thesis. The average improvement of PSNR at the target bit rates from 0.3bpp to 1.0bpp for the proposed scheme is 1.41dB and 4.09dB over the original SFS and the SPIHT algorithms respectively on two ultrasound test images, U1 and U7.

Moreover, the histogram initialization of the arithmetic coder is modified to suit the subbands' statistical distribution. The modified histogram is easier to learn and adapts to the actual coding sequence faster than the conventionally uniformly initialized histograms. Improved results of the compression rate on two sample subband images, image16 and 19, show the potential of modifying the histogram initialization for the context-based entropy coder.

7.2 Future work

The work done in this thesis can be extended along several interesting directions.

First, we could design further experiments to explore the reasons for the experimental evidence we have obtained to continue this thesis research work. For example, we have compared the GAP and MED based on the compression performance and prediction error. More experiments to explain the results are needed to explore which model should be used for what type of image. In particular, tests should be made using images with known statistical properties. For the issue of how many contexts to use for the context-based entropy coder, we have fixed this number to be a certain value for each sized subband. Further in depth analysis and research of these context sizes could be done to yield better compression ratios and to reduce the computational complexity.

The second improvement is making modifications on the parameter of context models, which are obtained using experiments in this thesis. If we use the two-dimensional linear prediction algorithm to adapt for each subband, the precision of predictors can be improved. It would be interesting to investigate possible methods to estimate these parameters in a more theoretical manner. Furthermore, the quantization of the predictors can also be adjusted to suit the statistical distribution of each subband instead of using a uniform quantizer. As we know, the improvement on contexts for each encoded symbol will enhance the efficiency of the conditional arithmetic coder, and accordingly will improve the optimization of the space-frequency representation of the images. Therefore, further research on the context modeling may achieve higher compression performance of the images.

The next enhancement would be to consider other areas in which the space-frequency segmentations can be improved. Chiu modifies Herley's original space frequency segmentation algorithm for ultrasound images. However, further improvements on the current method are still possible such as carefully choosing the quantizer sets and filters used. More experiments and analysis will optimize these parameters. The high computational cost in space-frequency segmented images should also be improved by possible solutions, for example investigating ways to limit searching the entire set of space-frequency segmentation and finding efficient cost metrics to simplify the calculation for basis choice.

Finally, modification of histogram initialization is a potential way to achieve better compression performance. Since the research conducted on histogram initialization in this thesis is only a preliminary attempt, further in depth analysis and experiments is expected to yield much better performance. Furthermore, histogram forgetting is another technique that can potentially improve the proposed context based entropy coder. Using this technique, a coefficient less than one is used to multiply the older histogram counts before it is incremented. The influence of the old symbols is gradually reduced. Entropy coders with modified histograms are more likely to have a higher compression ratio than the conventional coders with uniformly-initialized histograms.

Bibliography

[1] Anonymous, "Fast and efficient lossless image compression." *Proc. Data Compression Conf.* 1993,pp351-360

[2] A.Croisier, D.Esteban, and C. Galand. "Perfect Channel Splitting by Use of Interpolation/Decimation Techniques." *Proceedings Internatinal Conference on Information Science and Systems.* Piscataway, NJ: IEEE Press, 1976

[3] A.N.Netravili and B.G.Haskell, *Digital Pictures – Representation and Compression*. Applications of Communications Theory Series. Plenum Press. 1988

[4] A.Said and W.Pearlman, "A New Fast and Efficient Image Coder Based on Set Partitioning on Hierarchical Trees." *IEEE Trans. Circuits and Systems for Video Technology* , vol.6, pp.243-250, June 1996

[5] Alistair Moffat, "Arithmetic Coding Revisited.", *ACM Transactions on Information System*, vol.16,No.3, July 1998

[6] Anil K.Jain. "Image data compression: a review." *Proceedings of the IEEE*, 69(3):349-389, March 1981

[7] Arthur C.Fleischer, A.Everette James, Jr. " Introduction to Diagnostic Sonography " 1979

[8] Arthur C.Fleischer, A.Everette James, Jr. "Diagnostic Sonography, principles and clinical applications" 1989

[9] B.J.Erickson, A.Manduca, P.Palisson, K.R.Persons, D.Earnest, and V.Savcenko, "Wavelet Compression of Medical Images." *Radiology*, vol.206, pp.599-607,1998.

[10] Boliek, M.,Gormish, M.J., Schwartz, E.L.,and Keith, A. , " Next Generation Image Compression and Manipulation Using CREW", *Proc.IEEE ICIP,* 1997

[11] Buccigrossi, R., and Simoncelli, E.P. ,"EPWIC : Embedded Predictive Wavelet Image Coder", GRASP Laboratory, TR#414

[12] C.Herley, J.Kovacevic,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, pp3341-3349, December 1991.

[13] C. Herley, 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

[14] C.Sidney Burrus, Ramesh A.Gopinath, Haito Guo, *Introduction to Wavelets and Wavelet Transforms*, Prentic Hall, New Jersey 1998

[15] Carmen de Solà Fàbregas, Nguyen Phu Tri, “Ultrasound Image Coding using Shape-Adaptive DCT and Adaptive Quantizatin”, *Proceedings of SPIE*, pp.328-338, 1997

[16] Christos Chrysafis , Antonio Ortega "Efficient Context-Based Entropy Coding for Lossy Wavelet Image Compression " *Proc. Of 1997 Data Compression Conference*, pp.241-250, Mar. 1997

[17] Coifman, R.R. and Wickerhauser, M.V., “Entropy Based Algorithms for Best Basis Selection”, *IEEE Trans. Information Theory*, vol.38, no.2, Mar.1992, pp.713-718

[18] D.Taubman and A.Zakhor, "Multirate 3-D Subband Coding of Video" *IEEE Trans. Image Processing*, vol.3,pp.572-588, Sept.1994

[19] Darrel Hankerson, Greg A.Harris, Peter D.Johnson, Jr. “Introduction to Information Theory and Data compression”, 1998

[20] David Taubman, "High Performance Scalable Image Compression with EBCOT" *IEEE Transactions on Image Processing* ,vol.9, No.7, pp1158-1171, Jul.2000

[21] E. A. Riskin, "Optimum bit allocation via the generalized BFOS algorithm,"*IEEE Transactions on Information Theory*, vol. 37, pp.400-402, Mar. 1991

[22] Ed.Chiu, “Lossy Compression of Medical Ultrasound Images Using Space-Frequency Segmentation. ” *School of Engineering Science M.A.Sc. Thesis*: Simon Fraser University, 1999, http://www.ensc.sfu.ca/grad/theses/.

[23] Estaban, D. and Galand, C. “Application of Quadrature Mirror Filters to Split Band Voice Coding Schemes.” *Proc. ICASSP*, pp191-195 May1977.

[24] Farr, R.F., Allisy-Roberts,P.J. “Physics for medical imaging” London:Saunders, 1997.

[25] G.J.Sullivan "Efficient Scalar Quantization of Exponential and Laplacian Random Variables" *IEEE Trans. Information Theory* , vol.42, pp.1365-1374, September 1996

[26] G.K.Wallace “The JPEG still picture compression standard.”*Communications of the ACM*, 34(4):31-44,1991

[27] G.Langdon and C.Haidinyak, “Context-dependent distribution shaping and parameterization for lossless image compression.” *Appl. Dig.Image Proc.XVII, SPIE* , 1994,pp.62-70

[28] Gilbert Strang, Truong Nguyen, *Wavelets and Filger Banks*, Wellesley-Cambridge Press, 1996

[29] Hüseyin Abut , *Vector Quantization*, New York: IEEE Press, c1990

[30] J.B.O’Neal. “Predictive quantizing differential pulse code modulation for the transmission of television images.” *Bell Syst. Tech. J.*, 45(5):689-721. 1966

[31] J.M.Shapiro, "Embedded Image Coding Using Zerotrees of Wavelet Coefficients" *IEEE Trans. Signal Processing* , vol.41, pp.3445-3462, December 1993

[32] James Z. Wang, Gio Wiederhold, Jia Li, ``Wavelet-based progressive transmission and security filtering for medical image distribution,'' *Medical Image Databases*, S. Wong (ed.), Kluwer International Series in Engineering and Computer Science, Secs 465, 303-324 (Chapter 12), Kluwer, 1998

[33] Joseph Yeh, “Stressmark Model of Video Coding” July 1998 http://www.cs.berkeley.edu/~jyeh/liftview/node2.html.

[34] JPEG 2000 Committee “ JPEG 2000 requirements and profiles version 6.3”, ISO/IEC JTC 1/SC 29/WG 1 , July 2000 http://www.jpeg.org/public/wg1n1803.pdf

[35] K.Ramchandran 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

[36] Khalid Sayood, *Introduction to Data Compression*. Morgan Kauffman Publishers. Pp61-94, pp259-322,1996

[37] M.Antonini, M.Barlaud, P.Mathieu, and I.Daubechies, “Image coding using wavelet transform.” *IEEE Trans. Signal Processing*, vol.1, pp.205-221, Apr.1992.

[38] M.V.Wickhauser, “Acoustic signal compression with wavelet packets”, in *Wavelets: A Tutorial in Theory and Applications*, C.K.Chui, Ed. New York: Academic, 1992, pp.679-700

[39] Majid Rabbani and Paul W.Jones. *Digital Image Compression Techniques*, volume TT7 of *Tutorial Texts Series*. SPIE Optical Engineering Press. 1991

[40] Marcelo J.Weinberger, Cadiel Seroussi, and Guillermo Sapiro "LOCO-I : A Low Complexity, Context-Based, Lossless Image Compression Algorithm" *proc of IEEE Data Compression Conference*, pp.140-149, Snowbird, Utah, U.S.A. March 1996

[41] Marcelo J.Weinberger, Jorma J.Rissanen, and Ronald B.Arps "Applications of Universal Context Modeling to Lossless Compression of Gray-Scale Images" *IEEE Trans. Image Precessing*, Vol.5, No.4, April 1996

[42] Mark Nelson, “Arithmetic Coding + Statistical Modeling = Data Compression.” *Dr.Dobb’s Journal* , February 1991

[43] N.D Memonand K. Sayood, “Lossless image compression - a comparative study.” *Still Image Compression, SPIE Proc.*, vol.2418,pp8-20, 1995

[44] N.Jayant and P.Noll, *Digital Coding of Waveforms.* Englewood Cliffs, NJ:Prentice-Hall, pp154, 1984

[45] N. Memon and X. Wu, ``Recent Developments in Context-Based Predictive Techniques for Lossless Image Compression'', *The Computer Journal*, vol. 40, no. 2/3, p. 127-136, 1997.

[46] Petros A.Maragos, Ronald W.Schafer, Russell M.Mersereau, “Two-Dimensional Linear Prediction and its application to adaptive predictive coding of images”, *IEEE Transaction on Acoustics, Speech, and Signal processing*, pp1213-1229,vol..ASSP-32, No.6, Dec.1984.

[47] R.Coifman, Y.Meyer, S.Quake, and V.Wickerhauser, “Signal processing and compression with wave packets”, *Numerical Algorithms Research Group*, New Haven, CT: Yale University, 1990

[48] S.A. Martucci “Reversible compression of HDTV images using median adaptive prediction and arithmetic coding.”, *IEEE Int.Symp. onCircuits and Systems*, pp.1310-1313. IEEE Press, New York, 1990

[49] Shannon, C.E., “A Mathematical Theory of Communication”, *Bell System Technical Journal*, vol.27, no.3, pp.379-423,1948

[50] Subhasis Saha “Image Compression : From DCT to Wavelets – A Review.” University of California, Davis , March 2000, http://www.engr.ucdavis.edu/~ssaha/sahaimgcoding.html.

[51] T.Endoh and Y.Yamakazi “Progressive coding scheme for multilevel images”, *Proceedings of the Picture Coding Symposium*. Tokyo, pp21-22, SPIE, 1986

[52] Tsai, M.J., Villasenor, J.D., and Chen, F., “Stack-Run Image Coding”, *IEEE Trans. CSVT*, vol.6, no.5, Oct.1996, pp.519-521

[53] Vetterli M. and J.Kovacevic, *Wavelets and Subband Coding.* Englewood Cliffs, N.J.:Pretice_-Hall, 1995

[54] W.Pennebaker and J.Mitchell, *JPEG Still Image Data Compression Standard*. New York: Van Nostrand Reinhold, 1994

[55] Weidong Kou “Digital Image Compression, Algorithms and Standards”,pp149 1995

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

[57] 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

[58] X. Wu and N. Memon, ``Context-based, Adaptive, Lossless Image Codec'', *IEEE Trans. on Communications*, vol. 45, no. 4, p. 437-444, April 1997.

[59] Xiaolin Wu "Lossless Compression of Continuous-Tone Images via Context Selection, Quantization, and Modeling " *IEEE Trans. Image processing* , vol.6, pp.656-664, May 1997

[60] Xiaolin Wu , J.Chen "Context-modeling and entropy coding of wavelet coefficients for image compression " *Proc.ICSSP 97* Munich, Germany, pp. 3097-3100, 1997

[61] Xiaolin Wu “An Algorithmic Study on Lossless Image Compression”

[62] Youngjun Yoo, Antonio Ortega, Bin Yu "Image Subband Coding Using Context-Based Classification and Adaptive Quantization" *IEEE Trans. Image Processing* , Vol.8, No.12, pp.1702-1715, December 1999

[63] Zixiang Xiong, Kannan Ramchandran, Michael T.Orchard, “Space-Frequency Quantization for Wavelet Image Coding”, *IEEE Trans.IP*, vol.6,no.5,May 1997, pp677-693

Table A.1 : List of the experiment subbands

subband	name	depth	frequency	size
img1	barb_l2_sub1	2	low	128x128
img2	barb_l2_sub2	2	high	128x128
img3	barb_l2_sub3	2	high	128x128
img4	barb_l2_sub4	2	high	128x128
img5	barb_l3_sub1	3	low	64x64
img6	barb_l3_sub2	3	high	64x64
img7	barb_l3_sub3	3	high	64x64
img8	barb_l3_sub4	3	high	64x64
img9	barb_l3_sub5	3	high	64x64
img10	barb_l3_sub6	3	high	64x64
img11	barb_l3_sub7	3	high	64x64
img12	barb_l4_sub5	4	high	64x64
img13	barb_l5_sub5	5	high	64x64
img14	barb_l5_sub6	5	high	64x64
img15	barb_l5_sub7	5	high	64x64
img16	barb_l4_sub1	4	low	32x32
img17	barb_l4_sub2	4	high	32x32
img18	barb_l4_sub3	4	high	32x32
img19	barb_l4_sub4	4	high	32x32
img20	barb_l5_sub1	5	high	32x32
img21	barb_l5_sub2	5	high	32x32
img22	barb_l5_sub3	5	high	32x32
img23	barb_l5_sub4	5	high	32x32
img24	barb_l5_sub8	5	high	16x16
img25	barb_l5_sub9	5	high	16x16
img26	u1_l2_sub1	2	low	160x120
img27	u1_l2_sub2	2	high	160x120
img28	u1_l2_sub3	2	high	160x120
img29	u1_l2_sub4	2	high	160x120
img30	u1_l5_sub13	5	high	80x60

img31	u1_l4_sub5	4	high	80x60
img32	u1_l3_sub1	3	low	80x60
img33	u1_l3_sub2	3	high	80x60
img34	u1_l3_sub3	3	high	80x60
img35	u1_l3_sub4	3	high	80x60
img36	u1_l3_sub5	3	high	80x60
img37	u1_l3_sub6	3	high	80x60
img38	u1_l3_sub7	3	high	80x60
img39	u1_l5_sub10	5	high	40x30
img40	u1_l5_sub11	5	high	40x30
img41	u1_l5_sub12	5	high	40x30
img42	u1_l4_sub1	4	low	40x30
img43	u1_l4_sub2	4	high	40x30
img44	u1_l4_sub3	4	high	40x30
img45	u1_l4_sub4	4	high	40x30
img46	u1_l5_sub1	5	low	20x15
img47	u1_l5_sub2	5	high	20x15
img48	u1_l5_sub3	5	high	20x15
img49	u1_l5_sub4	5	high	20x15
img50	u1_l5_sub5	5	high	20x15
img51	u1_l5_sub6	5	high	20x15
img52	u1_l5_sub7	5	high	20x15
img53	u1_l5_sub8	5	high	20x15
img54	u1_l5_sub9	5	high	20x15

Table A.2: Compression results for different causal context coders

image			SP1		SP2		GAP		MED	
subband	size	frequency	bpp	d	bpp	d	bpp	d	bpp	d
img1	128x128	low	2.7835	10	2.5365	10	**2.3692**	10	2.3742	10
img2	128x128	high	0.3775	3	0.3804	7	**0.3578**	9	0.3666	3
img3	128x128	high	0.8786	7	0.9042	7	**0.8177**	10	0.8178	7
img4	128x128	high	0.1691	3	0.1580	3	**0.1550**	9	0.1667	3
img5	64x64	low	4.9708	6	4.6735	8	4.4782	10	**4.4253**	10
img6	64x64	high	1.6839	9	1.6763	6	1.6858	9	**1.6748**	9
img7	64x64	high	2.4384	2	2.4268	7	**2.3395**	10	2.3765	9
img8	64x64	high	1.3377	10	1.3455	10	1.3424	8	**1.3189**	9
img9	64x64	high	**0.0395**	3	0.0428	1	0.0420	4	0.0426	3
img10	64x64	high	**0.0734**	3	0.0791	3	0.0789	3	0.0790	3
img11	64x64	high	0.9815	5	0.9529	5	**0.9429**	9	0.9645	5
img12	64x64	high	1.6839	9	1.6763	6	1.6858	9	**1.6748**	9
img13	64x64	high	1.3645	9	1.3470	9	**1.3057**	9	1.3105	8
img14	64x64	high	2.4384	2	2.4268	7	**2.3395**	10	2.3765	9
img15	64x64	high	1.3377	10	1.3455	10	1.3424	8	**1.3189**	9
img16	32x32	low	5.9055	2	5.7441	3	5.6575	4	**5.5806**	4
img17	32x32	high	3.5687	1	3.5687	1	3.5687	1	3.5687	1
img18	32x32	high	4.3590	1	4.3589	1	**4.3537**	3	4.3589	1
img19	32x32	high	3.0027	1	3.0027	1	3.0027	1	3.0027	1
img20	32x32	high	0.5535	1	0.5535	1	0.5535	1	0.5535	1
img21	32x32	high	1.0331	5	1.0809	3	1.0486	3	**1.0468**	5
img22	32x32	high	0.1479	1	0.1479	1	0.1479	1	0.1479	1
img23	32x32	high	0.2621	2	0.2610	3	0.2618	3	**0.2575**	2
img24	16x16	high	0.1874	1	0.1874	1	0.1874	1	0.1874	1
img25	16x16	high	0.0781	1	0.0781	1	0.0781	1	0.0781	1
img26	160x120	low	2.4868	8	2.4416	10	2.4164	8	**2.4097**	10
img27	160x120	high	0.5852	6	0.5832	10	**0.5786**	6	0.5810	6
img28	160x120	high	0.3939	8	0.4352	10	**0.3682**	9	0.3743	8
img29	160x120	high	0.1892	9	0.2073	1	0.1830	10	**0.1798**	9
img30	80x60	high	0.0050	1	0.0050	1	0.0050	1	0.0050	1

img31	80x60	high	0.2164	1	0.2164	1	0.2164	1	0.2164	1
img32	80x60	low	3.5363	10	3.5282	8	**3.3815**	10	3.3867	10
img33	80x60	high	**2.7844**	10	2.8476	8	2.8130	10	2.8065	10
img34	80x60	high	1.1760	6	1.1662	10	1.1722	9	**1.1461**	9
img35	80x60	high	0.5374	9	0.5292	9	0.5257	10	**0.5217**	9
img36	80x60	high	**1.1251**	10	1.2027	9	1.1771	5	1.1511	10
img37	80x60	high	0.0350	1	0.0350	1	0.0350	1	0.0350	1
img38	80x60	high	0.1636	1	0.1636	1	0.1636	1	0.1636	1
img39	40x30	high	**0.7377**	7	0.9171	7	1.0665	7	0.8263	7
img40	40x30	high	0.0185	1	0.0185	1	0.0185	1	0.0185	1
img41	40x30	high	0.0185	1	0.0185	1	0.0185	1	0.0185	1
img42	40x30	low	1.0345	9	1.0490	8	1.0204	10	**0.9516**	8
img43	40x30	high	2.1311	9	2.0018	9	1.8091	9	**1.7700**	9
img44	40x30	high	1.8351	1	1.8351	1	1.8351	1	1.8351	1
img45	40x30	high	0.6604	1	0.6604	1	0.6604	1	0.6604	1
img46	20x15	low	**1.3192**	8	1.4189	4	1.4432	4	1.4300	7
img47	20x15	high	1.2198	3	1.2257	2	1.1333	10	**1.0989**	6
img48	20x15	high	0.9285	8	0.9208	2	0.9401	2	**0.7797**	7
img49	20x15	high	0.9806	8	0.8967	2	1.0065	8	**0.8655**	7
img50	20x15	high	0.3033	1	0.3033	1	0.3033	1	0.3033	1
img51	20x15	high	0.6302	6	0.5490	10	0.4117	10	**0.3685**	6
img52	20x15	high	3.2157	10	3.2315	10	2.8577	10	**2.7224**	10
img53	20x15	high	**1.1891**	10	1.3729	1	1.3069	10	1.2321	10
img54	20x15	high	**1.8160**	8	1.8991	9	1.8965	10	1.9305	9

Note :
bpp - bit per pixel
d - number of context status
SP1 - Simple Predictor 1
SP2 - Simple Predictor 2
GAP - Gradient Adjusted Prediction
MED - Median Edge Detection predictor

Table A.3 : Compression result for CSQM algorithm

Subband	Size	Frequency	Rate1	Rate2	Rate Diff.	Rate Diff%	Dist. Diff%	Opt nCont
img1	128x128	low	4.0961	4.3482	0.2521	6%	133.51%	2
img2	128x128	high	0.3371	0.4528	0.1157	34%	82.58%	4
img3	128x128	high	0.7745	0.9571	0.1826	24%	125.83%	5
img4	128x128	high	0.1482	0.2017	0.0535	36%	54.53%	3
img5	64x64	low	5.7685	6.0314	0.2629	5%	152.75%	1
img6	64x64	high	1.6031	1.8463	0.2432	15%	120.04%	3
img7	64x64	high	2.3413	2.5980	0.2567	11%	132.68%	3
img8	64x64	high	1.2175	1.4424	0.2249	18%	122.60%	3
img9	64x64	high	0.0466	0.0751	0.0285	61%	26.46%	1
img10	64x64	high	0.0878	0.1268	0.0390	44%	35.20%	2
img11	64x64	high	0.9060	1.1101	0.2041	23%	118.02%	2
img12	64x64	high	1.6031	1.8463	0.2432	15%	120.04%	3
img13	64x64	high	1.2819	1.5072	0.2253	18%	116.00%	3
img14	64x64	high	2.3413	2.5980	0.2567	11%	132.68%	3
img15	64x64	high	1.2175	1.4424	0.2249	18%	122.60%	2
img16	32x32	low	6.5250	6.8138	0.2888	4%	176.80%	1
img17	32x32	high	3.6498	3.9387	0.2889	8%	147.68%	1
img18	32x32	high	4.4977	4.7883	0.2906	6%	131.95%	1
img19	32x32	high	3.0210	3.3081	0.2871	10%	144.35%	1
img20	32x32	high	0.5798	0.7659	0.1861	32%	86.26%	2
img21	32x32	high	1.1407	1.3638	0.2231	20%	100.62%	2
img22	32x32	high	0.1731	0.2515	0.0784	45%	17.39%	1
img23	32x32	high	0.2836	0.3948	0.1112	39%	48.71%	2
img24	16x16	high	0.3058	0.4675	0.1617	53%	79.48%	1
img25	16x16	high	0.2083	0.3467	0.1384	66%	0.00%	1
img26	160x120	low	3.2400	3.4756	0.2356	7%	148.66%	5
img27	160x120	high	0.5104	0.6327	0.1223	24%	78.44%	5
img28	160x120	high	0.2953	0.3712	0.0759	26%	84.72%	3
img29	160x120	high	0.1426	0.1903	0.0477	33%	116.62%	3
img30	80x60	high	0.0138	0.0264	0.0126	91%	0.00%	1

img31	80x60	high	0.1978	0.2763	0.0785	40%	108.78%	2
img32	80x60	low	4.4002	4.6595	0.2593	6%	145.09%	5
img33	80x60	high	2.6687	2.9254	0.2593	6%	163.73%	3
img34	80x60	high	1.0768	1.2769	0.2001	19%	132.99%	3
img35	80x60	high	0.5035	0.6176	0.1141	23%	60.39%	3
img36	80x60	high	1.0049	1.1963	0.1914	19%	170.13%	3
img37	80x60	high	0.0490	0.0740	0.0250	51%	60.79%	1
img38	80x60	high	0.1563	0.2257	0.0694	44%	40.68%	1
img39	40x30	high	1.3804	1.6247	0.2443	18%	728.98%	3
img40	40x30	high	0.0500	0.0602	0.0102	20%	0.00%	1
img41	40x30	high	0.0500	0.0905	0.0405	81%	0.00%	1
img42	40x30	low	1.1708	1.3657	0.1949	17%	325.31%	3
img43	40x30	high	3.3508	3.6367	0.2859	9%	230.18%	2
img44	40x30	high	1.7417	2.0213	0.2796	16%	146.64%	2
img45	40x30	high	0.6531	0.8389	0.1858	28%	124.96%	2
img46	20x15	low	1.5115	1.6655	0.1540	10%	172.92%	2
img47	20x15	high	1.4563	1.6700	0.2137	15%	197.56%	2
img48	20x15	high	2.3864	2.7486	0.3622	15%	457.31%	2
img49	20x15	high	1.4942	1.6274	0.1332	9%	161.46%	2
img50	20x15	high	0.4389	0.5191	0.0802	18%	286.85%	1
img51	20x15	high	2.0631	2.1055	0.0424	2%	0.00%	1
img52	20x15	high	5.2225	5.5615	0.3390	6%	144.02%	1
img53	20x15	high	1.8483	2.1442	0.2959	16%	99.65%	2
img54	20x15	high	2.3713	2.7335	0.3622	15%	203.39%	2
					Average :	**24%**	**131.83%**	

Note:
Rate 1 - the bit rate when ignoring the distortion
Rate 2 - the bit rate when encoding the extra information to get rid of the truncation error
Rate Diff. - the more bit rate required to deal with the truncation distortion
Rate diff% - the percentage of more bit rate required comparing with Rate1
Dist. diff% - the percentage of more distortion generated by rate 1 when comparing with rate 2

Table A.4 : Compression result for DPCM on CSQM algorithm

Subband	Size	Frequency	Rate1	Rate2	Rate Diff.	Rate Diff%	Dist. Diff%	Opt nCont
img1	128x128	low	2.1194	2.3718	0.2524	12%	133.51%	5
img2	128x128	high	0.4694	0.5855	0.1161	25%	82.58%	5
img3	128x128	high	0.9161	1.0991	0.1830	20%	125.83%	3
img4	128x128	high	0.1902	0.2440	0.0538	28%	54.53%	3
img5	64x64	low	4.2098	4.4778	0.2680	6%	152.75%	2
img6	64x64	high	1.9385	2.1831	0.2582	10%	120.04%	3
img7	64x64	high	2.5886	2.8468	0.2446	13%	132.68%	3
img8	64x64	high	1.4956	1.7220	0.2264	15%	122.60%	3
img9	64x64	high	0.0704	0.1026	0.0404	29%	26.46%	3
img10	64x64	high	0.1400	0.1804	0.0152	78%	35.20%	2
img11	64x64	high	1.0922	1.2978	0.2056	19%	118.02%	3
img12	64x64	high	1.9385	2.1831	0.2446	13%	120.04%	3
img13	64x64	high	1.6032	1.8300	0.1812	17%	116.00%	3
img14	64x64	high	2.5886	2.8468	0.2253	16%	132.68%	3
img15	64x64	high	1.4956	1.7220	0.1067	24%	122.60%	3
img16	32x32	low	5.6376	5.9294	0.2918	5%	176.80%	1
img17	32x32	high	4.3264	4.6182	0.2918	7%	147.68%	1
img18	32x32	high	4.9353	5.2288	0.2935	6%	131.95%	1
img19	32x32	high	3.7732	4.0786	0.3054	8%	144.35%	2
img20	32x32	high	0.8585	1.0505	0.2923	5%	86.26%	2
img21	32x32	high	1.4347	1.6635	0.2882	10%	100.62%	2
img22	32x32	high	0.3070	0.3993	0.2882	9%	17.39%	2
img23	32x32	high	0.4567	0.5831	0.2994	13%	48.71%	2
img24	16x16	high	0.4735	0.6469	0.3594	21%	79.48%	1
img25	16x16	high	0.2669	0.4170	0.2517	24%	0.00%	1
img26	160x120	low	2.2856	2.5215	0.0265	27%	148.66%	5
img27	160x120	high	0.6475	0.7701	0.2443	12%	78.44%	5
img28	160x120	high	0.3550	0.4312	0.0102	16%	84.72%	3
img29	160x120	high	0.1885	0.2366	0.2517	11%	116.62%	3
img30	80x60	high	0.0169	0.0301	0.1540	7%	0.00%	1

img31	80x60	high	0.3127	0.3925	0.1814	11%	108.78%	2
img32	80x60	low	3.5107	3.7713	0.1723	9%	145.09%	3
img33	80x60	high	2.8713	3.1293	0.1331	7%	163.73%	3
img34	80x60	high	1.3716	1.5730	0.0801	12%	132.99%	3
img35	80x60	high	0.6744	0.7897	0.0424	2%	60.39%	3
img36	80x60	high	1.2433	1.4360	0.3489	7%	170.13%	2
img37	80x60	high	0.1166	0.1446	0.3159	14%	60.79%	3
img38	80x60	high	0.2559	0.3289	0.3367	13%	40.68%	2
img39	40x30	high	2.0171	2.2614	0.3367	13%	728.98%	3
img40	40x30	high	0.0625	0.0727	0.3367	13%	0.00%	1
img41	40x30	high	0.0625	0.0727	0.3367	13%	0.00%	1
img42	40x30	low	1.3731	1.5680	0.3367	13%	325.31%	2
img43	40x30	high	1.8816	2.1742	0.3367	13%	230.18%	3
img44	40x30	high	2.1505	2.4351	0.3367	13%	146.64%	2
img45	40x30	high	0.9525	1.1434	0.3367	13%	124.96%	2
img46	20x15	low	2.1148	2.2688	0.3367	13%	172.92%	2
img47	20x15	high	1.6857	1.8671	0.3367	13%	197.56%	1
img48	20x15	high	2.0027	2.1750	0.3367	13%	457.31%	2
img49	20x15	high	2.0228	2.1559	0.3367	13%	161.46%	2
img50	20x15	high	0.6634	0.7435	0.3367	13%	286.85%	1
img51	20x15	high	2.0741	2.1165	0.3367	13%	0.00%	2
img52	20x15	high	4.8801	5.2290	0.3367	13%	144.02%	1
img53	20x15	high	2.2675	2.5834	0.3367	13%	99.65%	2
img54	20x15	high	2.5576	2.8943	0.3367	13%	203.39%	1
					Average :	**15%**	**131.83%**	

Note:
Rate 1 - the bit rate when ignoring the distortion
Rate 2 - the bit rate when encoding the extra information to get rid of the truncation error
Rate Diff. - the more bit rate required to deal with the truncation distortion
Rate diff% - the percentage of more bit rate required comparing with Rate1
Dist. diff% - the percentage of more distortion generated by rate 1 when comparing with rate 2

Table A.5 : Compression result for multi-resulotion algorithm

Subband	Size	Frequency	Rate1_opt	Rate2_fix	Rate Diff.	Rate Diff%
img1	128x128	low	3.1500	3.1575	0.0075	0.2%
img2	128x128	high	0.3356	0.3364	0.0008	0.2%
img3	128x128	high	0.7436	0.7461	0.0025	0.3%
img4	128x128	high	0.1373	0.1375	0.0002	0.1%
img5	64x64	low	5.2173	5.2682	0.0509	1.0%
img6	64x64	high	1.6106	1.6396	0.0290	1.8%
img7	64x64	high	2.3139	2.3506	0.0367	1.6%
img8	64x64	high	1.2640	1.2796	0.0156	1.2%
img9	64x64	high	0.0492	0.0452	-0.0040	-8.8%
img10	64x64	high	0.0796	0.0798	0.0002	0.3%
img11	64x64	high	0.8935	0.9005	0.0070	0.8%
img12	64x64	high	1.6106	1.6396	0.0290	1.8%
img13	64x64	high	1.2861	1.3041	0.0180	1.4%
img14	64x64	high	2.3139	2.3506	0.0367	1.6%
img15	64x64	high	1.2640	1.2796	0.0156	1.2%
img16	32x32	low	6.4842	6.6847	0.2005	3.0%
img17	32x32	high	3.9374	4.1254	0.1880	4.6%
img18	32x32	high	4.7187	4.9366	0.2179	4.4%
img19	32x32	high	3.2492	3.4248	0.1756	5.1%
img20	32x32	high	0.5708	0.5923	0.0215	3.6%
img21	32x32	high	1.1454	1.1550	0.0096	0.8%
img22	32x32	high	0.1797	0.1820	0.0023	1.3%
img23	32x32	high	0.2833	0.2895	0.0062	2.1%
img24	16x16	high	0.3474	0.2911	-0.0563	-19.3%
img25	16x16	high	0.2085	0.1758	-0.0327	-18.6%
img26	160x120	low	2.9770	2.9920	0.0150	0.5%
img27	160x120	high	0.4987	0.5011	0.0024	0.5%
img28	160x120	high	0.2719	0.2717	-0.0002	-0.1%
img29	160x120	high	0.1358	0.1355	-0.0003	-0.2%
img30	80x60	high	0.0155	0.0138	-0.0017	-12.3%

img31	80x60	high	0.1920	0.1914	-0.0006	-0.3%
img32	80x60	low	3.9712	4.0109	0.0397	1.0%
img33	80x60	high	2.6744	2.7133	0.0389	1.4%
img34	80x60	high	1.0611	1.0726	0.0115	1.1%
img35	80x60	high	0.4926	0.4985	0.0059	1.2%
img36	80x60	high	0.9814	0.9862	0.0048	0.5%
img37	80x60	high	0.0457	0.0442	-0.0015	-3.4%
img38	80x60	high	0.1671	0.1667	-0.0004	-0.2%
img39	40x30	high	1.6537	1.6592	0.0055	0.3%
img40	40x30	high	0.0839	0.0894	0.0055	6.2%
img41	40x30	high	0.0538	0.0468	-0.0070	-15.0%
img42	40x30	low	1.5065	1.5120	0.0055	0.4%
img43	40x30	high	2.4399	2.4631	0.0232	0.9%
img44	40x30	high	1.8501	1.9271	0.0770	4.0%
img45	40x30	high	0.6815	0.7032	0.0217	3.1%
img46	20x15	low	2.0675	2.0819	0.0144	0.7%
img47	20x15	high	1.7607	1.7750	0.0143	0.8%
img48	20x15	high	1.6883	1.7027	0.0144	0.8%
img49	20x15	high	1.6828	1.6972	0.0144	0.8%
img50	20x15	high	0.6950	0.7093	0.0143	2.0%
img51	20x15	high	2.0526	2.0669	0.0143	0.7%
img52	20x15	high	4.6351	4.5643	-0.0708	-1.6%
img53	20x15	high	1.3123	1.3129	0.0006	0.0%
img54	20x15	high	1.6864	1.6805	-0.0059	-0.4%
					Average :	**-0.3%**

Note:
Rate1_opt - the bit rate with optimized context number
Rate2_fix - the bit rate with fixed context number (66)
Rate Diff. - the difference between the two rates (rate2 - rate1)
Rate diff% - the percentage of the rate difference

Table A.6 : Compression result for non-causal context coders

Subband	Size	Frequency	CSQM	DPCM on CSQM	Multi-res. Coder
img1	128x128	low	4.3482	**2.3718**	3.1575
img2	128x128	high	0.4528	0.5855	**0.3364**
img3	128x128	high	0.9571	1.0991	**0.7461**
img4	128x128	high	0.2017	0.2440	**0.1375**
img5	64x64	low	6.0314	**4.4778**	5.2682
img6	64x64	high	1.8463	2.1831	**1.6396**
img7	64x64	high	2.5980	2.8468	**2.3506**
img8	64x64	high	1.4424	1.7220	**1.2796**
img9	64x64	high	0.0751	0.1026	**0.0452**
img10	64x64	high	0.1268	0.1804	**0.0798**
img11	64x64	high	1.1101	1.2978	**0.9005**
img12	64x64	high	1.8463	2.1831	**1.6396**
img13	64x64	high	1.5072	1.8300	**1.3041**
img14	64x64	high	2.5980	2.8468	**2.3506**
img15	64x64	high	1.4424	1.7220	**1.2796**
img16	32x32	low	6.8138	**5.9294**	6.6847
img17	32x32	high	**3.9387**	4.6182	4.1254
img18	32x32	high	**4.7883**	5.2288	4.9366
img19	32x32	high	**3.3081**	4.0786	3.4248
img20	32x32	high	0.7659	1.0505	**0.5923**
img21	32x32	high	1.3638	1.6635	**1.155**
img22	32x32	high	0.2515	0.3993	**0.182**
img23	32x32	high	0.3948	0.5831	**0.2895**
img24	16x16	high	0.4675	0.6469	**0.2911**
img25	16x16	high	0.3467	0.4170	**0.1758**
img26	160x120	low	3.4756	**2.5215**	2.9920
img27	160x120	high	0.6327	0.7701	**0.5011**
img28	160x120	high	0.3712	0.4312	**0.2717**
img29	160x120	high	0.1903	0.2366	**0.1355**
img30	80x60	high	0.0264	0.0301	**0.0138**

img31	80x60	high	0.2763	0.3925	**0.1914**
img32	80x60	low	4.6595	**3.7713**	4.0109
img33	80x60	high	2.9254	3.1293	**2.7133**
img34	80x60	high	1.2769	1.5730	**1.0726**
img35	80x60	high	0.6176	0.7897	**0.4985**
img36	80x60	high	1.1963	1.4360	**0.9862**
img37	80x60	high	0.0740	0.1446	**0.0442**
img38	80x60	high	0.2257	0.3289	**0.1667**
img39	40x30	high	**1.6247**	2.2614	1.6592
img40	40x30	high	**0.0602**	0.0727	0.0894
img41	40x30	high	0.0905	0.0727	**0.0468**
img42	40x30	low	**1.3657**	1.5680	1.512
img43	40x30	high	3.6367	**2.1742**	2.4631
img44	40x30	high	2.0213	2.4351	**1.9271**
img45	40x30	high	0.8389	1.1434	**0.7032**
img46	20x15	low	**1.6655**	2.2688	2.0819
img47	20x15	high	**1.6700**	1.8671	1.775
img48	20x15	high	2.7486	2.1750	**1.7027**
img49	20x15	high	**1.6274**	2.1559	1.6972
img50	20x15	high	**0.5191**	0.7435	0.7093
img51	20x15	high	2.1055	2.1165	**2.0669**
img52	20x15	high	5.5615	5.2290	**4.5643**
img53	20x15	high	2.1442	2.5834	**1.3129**
img54	20x15	high	2.7335	2.8943	**1.6805**

Table A.7 : Compression result for different entropy coders

Subband	Size	Frequency	Basic Arith.	cauasal	non-causal
img1	128x128	low	4.0938	**2.3742**	3.1575
img2	128x128	high	0.4400	0.3666	**0.3364**
img3	128x128	high	1.0010	0.8178	**0.7461**
img4	128x128	high	0.2048	0.1667	**0.1375**
img5	64x64	low	5.6275	**4.4253**	5.2682
img6	64x64	high	1.7277	1.6748	**1.6396**
img7	64x64	high	2.4807	2.3765	**2.3506**
img8	64x64	high	1.3772	1.3189	**1.2796**
img9	64x64	high	**0.0413**	0.0426	0.0452
img10	64x64	high	0.0877	**0.0790**	0.0798
img11	64x64	high	1.0717	0.9645	**0.9005**
img12	64x64	high	1.7277	1.6748	**1.6396**
img13	64x64	high	1.4064	1.3105	**1.3041**
img14	64x64	high	2.4807	2.3765	**2.3506**
img15	64x64	high	1.3772	1.3189	**1.2796**
img16	32x32	low	6.0268	**5.5806**	6.6847
img17	32x32	high	**3.5629**	3.5687	4.1254
img18	32x32	high	**4.3531**	4.3589	4.9366
img19	32x32	high	**2.9968**	3.0027	3.4248
img20	32x32	high	**0.5477**	0.5535	0.5923
img21	32x32	high	1.0999	**1.0468**	1.1550
img22	32x32	high	**0.1420**	0.1479	0.1820
img23	32x32	high	0.2707	**0.2575**	0.2895
img24	16x16	high	**0.1640**	0.1874	0.2911
img25	16x16	high	**0.0547**	0.0781	0.1758
img26	160x120	low	3.7081	**2.4097**	2.9920
img27	160x120	high	0.6341	0.5810	**0.5011**
img28	160x120	high	0.4387	0.3743	**0.2717**
img29	160x120	high	0.2070	0.1798	**0.1355**
img30	80x60	high	**0.0038**	0.0050	0.0138

img31	80x60	high	0.2152	0.2164	**0.1914**
img32	80x60	low	4.5968	**3.3867**	4.0109
img33	80x60	high	2.8932	2.8065	**2.7133**
img34	80x60	high	1.2243	1.1461	**1.0726**
img35	80x60	high	0.5652	0.5217	**0.4985**
img36	80x60	high	1.2517	1.1511	**0.9862**
img37	80x60	high	**0.0338**	0.0350	0.0442
img38	80x60	high	**0.1623**	0.1636	0.1667
img39	40x30	high	1.5156	**0.8263**	1.6592
img40	40x30	high	**0.0135**	0.0185	0.0894
img41	40x30	high	**0.0135**	0.0185	0.0468
img42	40x30	low	1.3545	**0.9516**	1.5120
img43	40x30	high	3.2294	**1.7700**	2.4631
img44	40x30	high	**1.8301**	1.8351	1.9271
img45	40x30	high	**0.6554**	0.6604	0.7032
img46	20x15	low	1.5439	**1.4300**	2.0819
img47	20x15	high	1.2740	**1.0989**	1.7750
img48	20x15	high	1.1901	**0.7797**	1.7027
img49	20x15	high	1.1854	**0.8655**	1.6972
img50	20x15	high	**0.2833**	0.3033	0.7093
img51	20x15	high	1.4826	**0.3685**	2.0669
img52	20x15	high	4.3308	**2.7224**	4.5643
img53	20x15	high	1.3529	**1.2321**	1.3129
img54	20x15	high	1.9930	1.9305	**1.6805**

Note :
Coder 1 - the basic arithmetic coder without any context
Coder 2 - the causal context based entropy coder (MED)
Coder 3 - the non-causal context based entropp coder
Coder 4 - the CSQM algorithm
Coder 5 - the multirate coder (layered zero coding and layered PCM)

Appendix B

Ultrasound Physics

The piezoelectric or pressure-electric effect discovered by Pierre Curie has been applied to produce and detect ultrasonic waves. Piezoelectric crystals can convert electrical voltage into sound energy , and vice versa. They can produce voltage when subjected to the same frequency pressure.

Ultrasound is a longitudinal pressure wave composed of periodic compression and rarefaction of the surrounding material. It is produced by the transducer, occurring at a frequency above human hearing. Figure B.1 gives a graphical illustration of the ultrasonic waves.

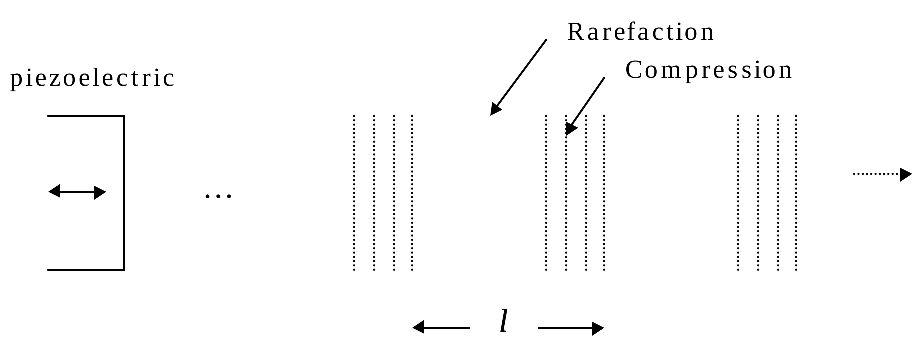


Figure B.1 – Production of ultrasound waves by a piezoelectric crystal

The above mode is called pulsed mode which is generated by a DC of few hundred volts. If an alternating (AC) voltage is applied, the transducer will work in continuous-wave mode to produce successive compression and rarefaction. Both the pulsed and continuous-wave modes are used in ultrasound imaging.

Three parameters are used to describe an ultrasound wave: the *wavelength* (l) , which includes one complete cycle of the compression and the rarefaction phase of the wave, the *frequency* (f) and the *velocity* (v) of wave propagation which is defined by the following equation :

$$v = lf \qquad \text{B.1}$$

Unlike light and X-rays, the velocity v of ultrasound depends on the material through which it travels, independent of its frequency. People often use the acoustic impedance (Z) to describe the property of the media. It is defined using the following equation :

$$Z = vr \qquad \text{B.2}$$

where r is the density of the material.

The acoustic impedance is used to define the reflected fraction R between two different materials . The reflected energy of an ultrasound wave at the interface of two different media (in Figure A.1) depends on not only incidence angle, but the acoustic impedance of two media as well.

When the sound wave strikes the surface at or nearly at right angles, the reflected fraction is :

$$R = \frac{(Z_1 - Z_2)^2}{(Z_1 + Z_2)^2} \qquad \text{B.3}$$

Suppose we have an angle of incidence (q_i), and an angle of refraction (q_r) as shown in Figure A.1, the ratio of two velocities of sound in two different materials equals to :

$$\frac{v_1}{v_2} = \frac{\sin q_i}{\sin q_r} \qquad \text{B.4}$$

This is called Snell’s law applied in the situation that a beam hits a large smooth interface between two different media.

When the surface of the tissue is not ideally smooth, and has ripples equal to approximately a wavelength, the reflected beams spread out over an angle. Consequently, the transducer will receive a wrong echo signal at that incident angle. This is called the diffuse reflection, causing speckles in ultrasound images.

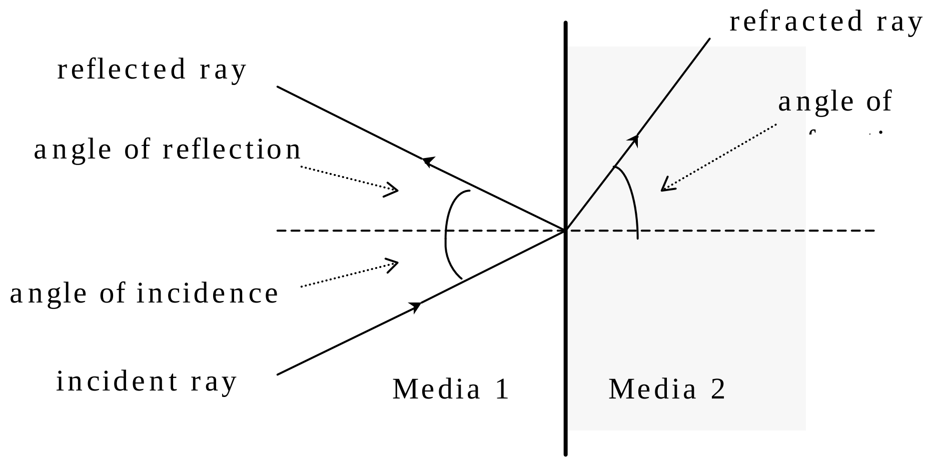


Figure B.2 – Reflection and refraction of sound waves

This basic background knowledge of ultrasound physics helps reader to understand the distinctive features of ultrasound images, and the way to compress them accordingly.

Appendix C

Quantizer Sets for Ultrasound Images, 0.5bpp

HP: high pass subbands

SO: space-only subbands

LP: low pass subbands

	Step size	Alphabet size	Minimum output
HP	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

	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
S	30.3	10	-128.0
O	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
	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
LP	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
	14.0	64	-448.0
	12.0	64	-384.0

Appendix D

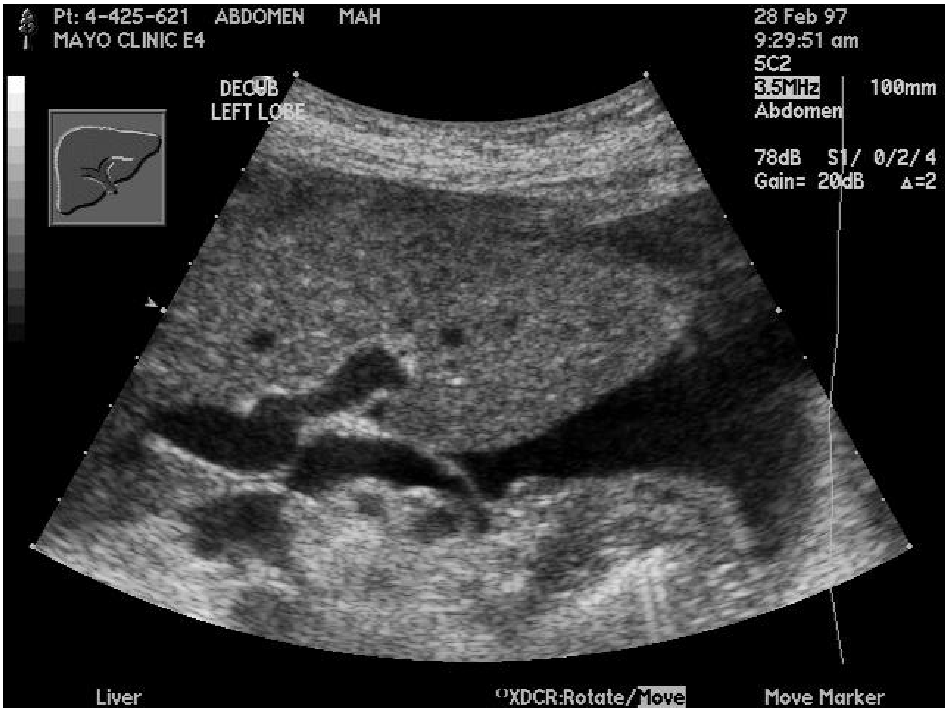
Images in the Experiments

Barbara (512x512)

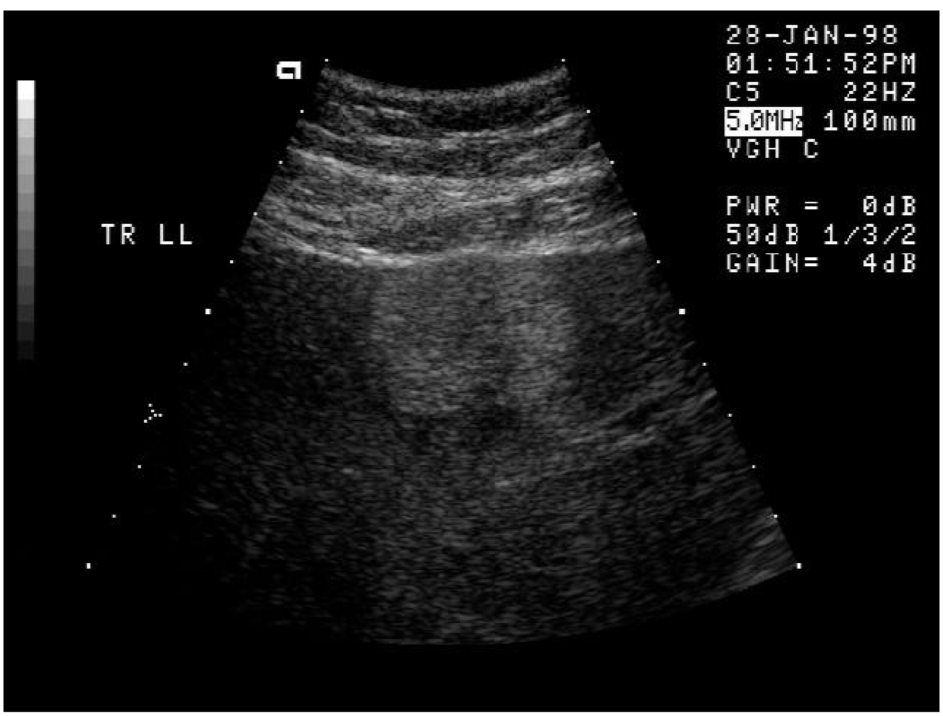
Airplane (512x512)

Bridge (512x512)

U1 (640x480)



U7 (640x480)



t11 (640x480)

t12 (640x480)

Appendix E

Ultrasound-only Images (192x256)

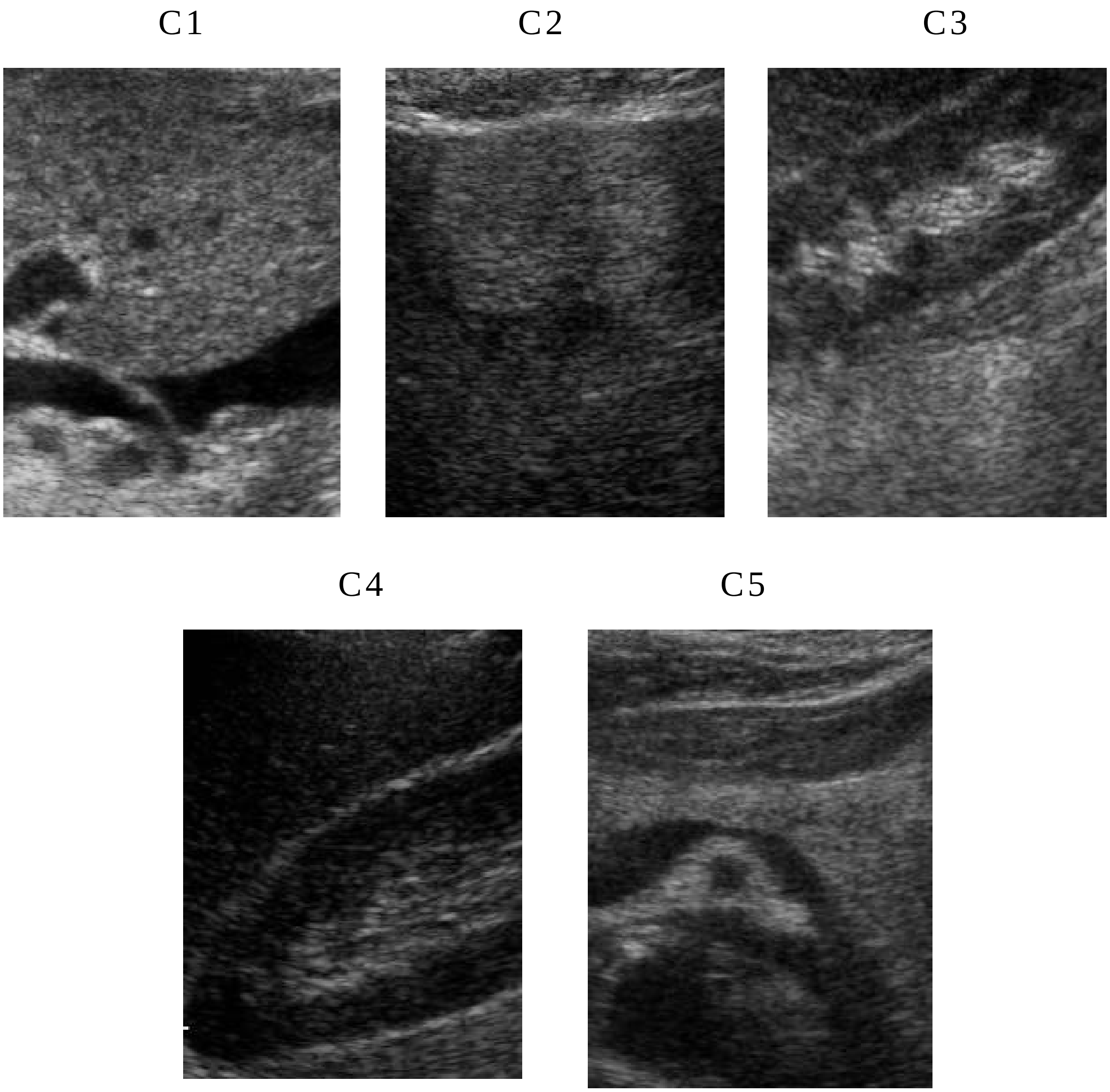


Figure E.1 – Original C1 to C5 images

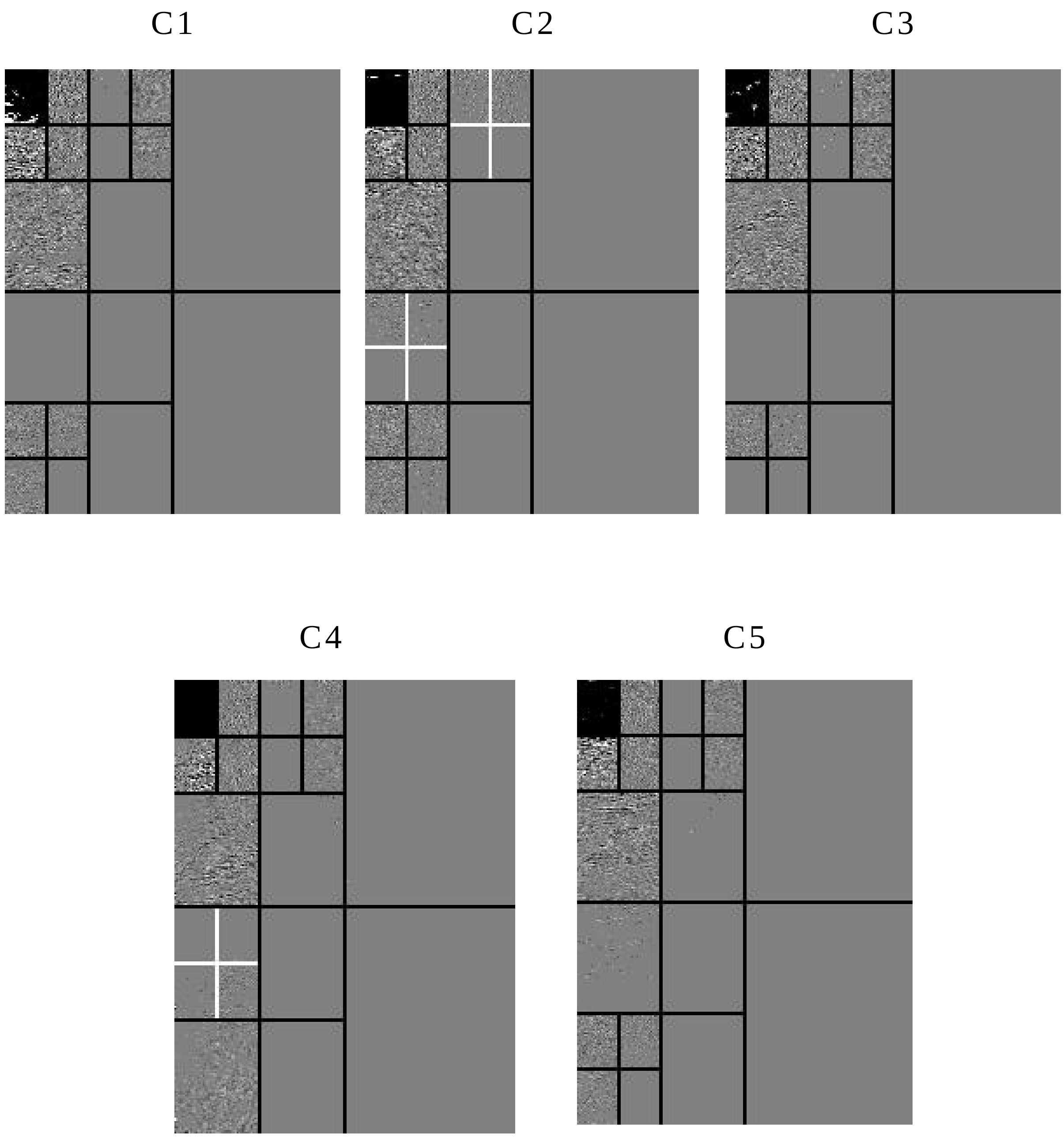


Figure E.2 – The SFS partitions, 0.5bpp

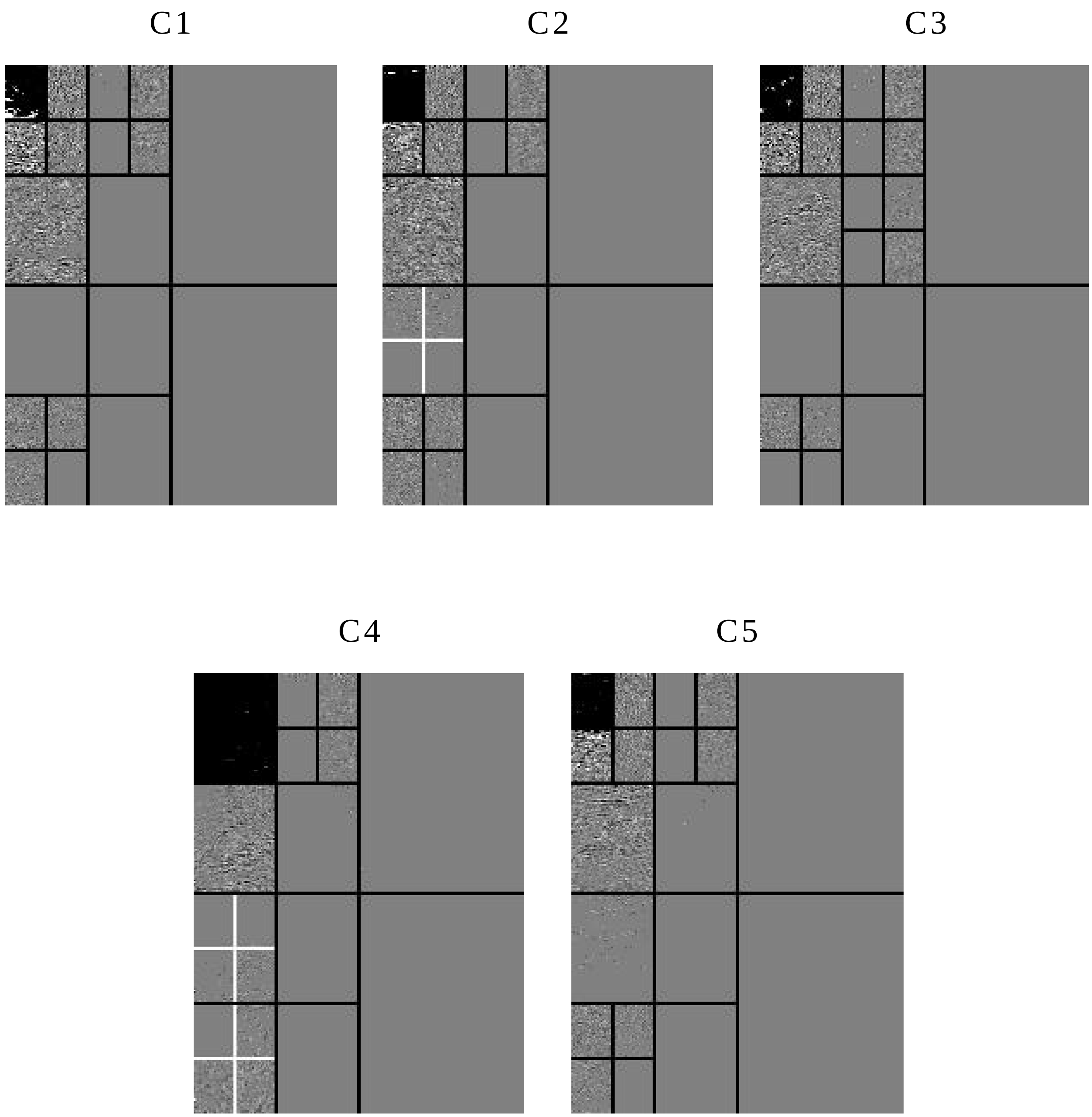


Figure E.3 – The CBEC partitions, 0.5bpp