



National Library
of Canada

Bibliothèque nationale
du Canada

Acquisitions and
Bibliographic Services Branch

Direction des acquisitions et
des services bibliographiques

395 Wellington Street
Ottawa, Ontario
K1A 0N4

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file *Votre référence*

Our file *Notre référence*

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

ON THE INTEGRATION OF MOTION COMPENSATION WITH SUBBAND FILTER CODING TECHNIQUES

by

Marlo Rene Gothe

B.A.Sc. (Eng.Sc.) Simon Fraser University, 1991

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

© Marlo Rene Gothe 1993
SIMON FRASER UNIVERSITY
July 7, 1993

All rights reserved. This work may not be
reproduced in whole or in part, by photocopy
or other means, without the permission of the author.



National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file *Votre référence*

Our file *Notre référence*

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-91170-0

Canada

APPROVAL

Name: Marlo Rene Gothe
Degree: Master of Applied Science
Title of thesis : On The Integration of Motion Compensation With Sub-band Filter Coding Techniques

Examining Committee: Dr. John Jones
Associate Professor, Engineering Science, SFU
Graduate Chair

Dr. Jacques Vaisey
Assistant Professor, Engineering Science, SFU
Senior Supervisor

~~Dr.~~ John S. Bird
Associate Professor, Engineering Science, SFU
Internal Supervisor

~~Mr. Elliot~~ Freedman
Advanced Technology Group, MPR Teltech LTD.
External ~~Committee~~ Member

Dr. Paul Ho
Associate Professor, Engineering Science, SFU
Examiner

Date Approved: July 7, 1993

PARTIAL COPYRIGHT LICENSE

I hereby grant to Simon Fraser University the right to lend my thesis, project or extended essay (the title of which is shown below) to users of the Simon Fraser University Library, and to make partial or single copies only for such users or in response to a request from the library of any other university, or other educational institution, on its own behalf or for one of its users. I further agree that permission for multiple copying of this work for scholarly purposes may be granted by me or the Dean of Graduate Studies. It is understood that copying or publication of this work for financial gain shall not be allowed without my written permission.

Title of Thesis/Project/Extended Essay

"On the Integration of Motion Compensation with Subband Filter Coding Techniques"

Author:

(signature)

Marlo R. GOTHE

(name)

July 7, 1993

(date)

Abstract

This thesis presents an empirical study of digital video compression source encoders and decoders. More specifically, codecs that integrate motion compensation with subband filters were studied. Motion compensation (MC) removes temporal correlation from video information and, as a result, there is a significant reduction in the codecs' bit rate. Current industry coding standards cascade MC with the discrete cosine transform (DCT), but visual blocking impairments, especially at low data rates, are undesirable and this system is not easily divisible into a multiresolution signal. Because of this, alternative coding methods integrated with MC need to be investigated. Subband filtering is one possible alternative method. This technique performs a frequency decomposition of a source; in video, it can be done both spatially in the image plane and temporally between frames. The major benefit is its ability to compact source energy into a small number of frequency bands.

The performance of twenty-two codecs was studied. The simplest codecs include configurations such as PCM, DPCM, MC, MC-DCT, spatial subband filtering, and temporal subband filtering. The remaining, more complex, codecs consisted of combinations of pair and triple orderings of MC, spatial subband filtering, and temporal subband filtering. Simulations were performed assuming an error free communications channel and using three standard source test video sequences. For all systems, uniform quantizers followed by zeroth-order-entropy measurements were used to represent a generic codec.

The results show that, for high motion video, a multiresolution spatial subband filter bank followed by MC has comparable performance to an MC-DCT coder, and that, for low motion video, temporal-spatial subband filter banks followed by MC performed better than an MC-DCT system. In addition, short kernel subband filter sets performed best. These conclusions would be useful to a video codec designer.

Acknowledgements

I would like to thank my senior supervisor, Dr. Jacques Vaisey, for his support and guidance throughout this thesis project.

I am grateful to Canadian Cable Labs, the Advance Systems Institute, and SFU Graduate Fellowship Committee for their financial support through research grants, scholarships, and fellowships respectively.

Contents

Abstract	iii
Acknowledgements	iv
1 Introduction	1
2 Background of Video Coding Tools	7
2.1 Sampling	7
2.2 Quantization	9
2.3 PCM and DPCM	13
2.4 Motion Compensation	16
2.5 Transform Coding	20
2.6 Run-length and Entropy Coding	21
2.7 Subband and Pyramidal Coding	22
3 Performance and Statistical Measures	28
3.1 Video Performance Measures	28
3.2 Video Statistical Measures	29
4 MC - Subband Filtering Video Codec Design	31
4.1 Video Sequence Notation	33
4.2 MC and Subband Filtering Codecs	33
4.2.1 Codecs Using One Video Coding Tool	34
4.2.2 Codecs Using Two Video Coding Tools	36
4.2.3 Codecs Using Three Video Coding Tools	39
4.3 Expected Codec Performance Rankings	41

4.4	Uniform Quantizer Design Methods	43
4.5	Bit Allocation Methods	44
4.5.1	A Filter Bank Noise Power Weighting Estimate	46
5	Video Codec Simulations and Results	49
5.1	Test Video Sequences	49
5.2	Subband Filtering Filter Sets	50
5.3	Simulations and Results	52
5.3.1	Video Codec Results for Group 1: PCM, DPCM, M, and MDCT	54
5.3.2	Video Codec Results for Group 2: S, T, TS, MS, and MT . .	57
5.3.3	Video Codec Results for Group 3: SM, SM1, TM, and TM1 .	69
5.3.4	Video Codec Results for Group 4: TSM, TSM1, MTS, SMT, and TMS	77
5.3.5	Video Codec Results Comparison of All Systems	81
5.3.6	Subjective Quality Evaluations of Best Systems	89
6	Conclusions	92
A	Video Compression Chip Sets	95
B	Seven Filter Impulse Responses	96
	Bibliography	101

List of Tables

5.1	One-Dimensional Weighting Factors for the Seven Filter Sets	51
5.2	Two-Dimensional Weighting Factors for the Seven Filter Sets	53
5.3	Group 1 Correlation and Weighted Variance Video Statistics	55
5.4	Group 2 Correlation and Weighted Variance Video Statistics	68
5.5	Group 3 Correlation and Weighted Variance Video Statistics	77
5.6	Group 4 Correlation and Weighted Variance Video Statistics	80
A.1	Nine Chip Set Manufactures	95
B.1	Smith and Barnwell (1986) 2 Tap QMF Impulse Response	96
B.2	LeGall and Tabatabai (1988) 3-5 Tap PRF Impulse Response	96
B.3	LeGall and Tabatabai (1988) 4 tap PRF Impulse Response	97
B.4	Smith and Barnwell (1986) 8 Tap CQF Impulse Response	97
B.5	Smith and Barnwell (1986) 16 Tap CQF Impulse Response	97
B.6	Johnston (1980) 16b tap QMF Impulse Response	98
B.7	Johnston (1980) 32c Tap QMF Impulse Response	98

List of Figures

2.1	Quantizer Block Diagram and Noise Model	10
2.2	Many-to-one Scalar Quantizer Mapping	10
2.3	Midtread Seven Level Uniform Quantizer Transfer Function	12
2.4	Block Diagram of a DPCM Coder and Decoder	14
2.5	Spatial Prediction Pixel Orientation	15
2.6	Block Motion Compensation	17
2.7	A Motion Compensation Codec	19
2.8	Subband Analysis/Synthesis Filter Structure	23
2.9	Analysis Filter Set Frequency Response	24
2.10	Spatial Analysis Subband Filtering	25
2.11	Spatial Synthesis Subband Filtering	25
2.12	Block Diagram of a Pyramidal Codec	26
4.1	A Symmetric Video Codec Configuration	31
4.2	A Video Codec with Feedback	32
4.3	The Coding of a Video Sequence using Motion Compensation	35
4.4	A Temporally Filtered Video Sequence	36
4.5	An MC - Temporally Filtered Video Sequence	40
4.6	An MC DFD Frame and PDF From the "Ping Pong" Video Sequence	43
4.7	Uniform Quantizer a.) Distortion versus Step-size and b.) Distortion versus Bit Rate Relationships	44
4.8	BFOS Distortion versus Rate Relationship	46
4.9	A Subband Synthesis Filter	46
4.10	Spectral Imaging of Interpolation Operator	47

5.1	First Frames of Simulation Test Video Sequences <i>missa</i> , <i>sales</i> , and <i>pongi</i>	50
5.2	The Seven Filter Sets Frequency Responses	52
5.3	PSNR versus Entropy for Group 1 Systems: PCM, DPCM, M, and M-DCT	56
5.4	PSNR versus Entropy for the S System	59
5.5	PSNR versus Entropy for the T System	61
5.6	PSNR versus Entropy for the TS System with the 2-QMF Temporal Filter Set	62
5.7	PSNR versus Entropy for the TS System with the 3-5-PRF Temporal Filter Set	63
5.8	PSNR versus Entropy for the TS System with the 8-CQF Temporal Filter Set	64
5.9	PSNR versus Entropy for the MS and MT Systems	66
5.10	PSNR versus Entropy for Group 2 Systems: S, T, TS, MS and MT	67
5.11	PSNR versus Entropy for the SM System	71
5.12	PSNR versus Entropy for the SM1 System	72
5.13	PSNR versus Entropy for the TM System	73
5.14	PSNR versus Entropy for the TM1 System	75
5.15	PSNR versus Entropy for Group 3 Systems: SM, SM1, TM, and TM1	76
5.16	PSNR versus Entropy for Group 4 Systems: TSM, TSM1, MTS, SMT, and TMS	79
5.17	PSNR versus Entropy for All Systems, <i>pongi</i> Sequence	82
5.18	PSNR versus Entropy for All Systems, <i>missa</i> Sequence	83
5.19	PSNR versus Entropy for All Systems, <i>sales</i> Sequence	84
5.20	PSNR versus Entropy for the Best Performing Systems	85
5.21	An MDCT Codec Block Diagram	86
5.22	An SM Codec Block Diagram	87
5.23	A TSM Codec Block Diagram	88
5.24	Subjective Comparisons at Four Different Bit Rates for the <i>pongi</i> Sequence	90
5.25	Subjective Comparisons at Four Different Bit Rates for the <i>missa</i> Sequence	91

Chapter 1

Introduction

Among the reasons for the growth and interest in video communications research is the role played by improved technologies and industry-driven video product developments. In North America, the development of a new television format called high definition television (HDTV) has resulted in much funding for video research and, as the standard becomes defined, new industries will be created for the installation and upkeep of this new video service. At present, corporate video conferencing services have been developed for specialized communication needs and, in the future, the mass market will have access to video phones for day to day communications.

Like most industries, the communications industry is confined to the time/cost constraints of existing communications plants. In the cable television industry, the plants have a limited bandwidth, which restricts the number of television channels and other information services it can deliver to its users. In the satellite industry, the amount of radio frequency bandwidth available for new uses is finite. Having to work within these constraints, the communications industry continues to strive for an increase in the efficiency of its existing facilities to provide new and improved services to its users.

There are two areas within which the efficiency of existing plants may be increased: channel communications and source communications. Channel communications deals with methods that increase the efficiency of the channel within given constraints, whereas source communications focuses on methods that remove redundancy from a source before the data items are sent to the channel for transmission. Most practical systems employ both source and channel coding.

Video communications falls mostly within the source communications classification. Video coders/decoders (codecs) use signal processing techniques to remove redundancy from video sequences before supplying the channel with a data stream. This takes place if there are no channel communication errors. For systems which operate on channels with non-zero error rates, the integration of both source and channel coding methods can help increase the overall system performance (Vaisey, Yuen, and Cavers 1992).

Several video source coding standards have been developed and include JPEG¹ (1990), MPEG² (LeGall 1991), and H.261 (Liou 1991). JPEG is a standard for coding still images and, for example, can provide good quality compression of grayscale images at an average coding bit rate of 0.7 bits/pixel. Although designed for coding still images, JPEG is also used in video compression. Among the advantages for its use in video compression are the symmetry between its coder and decoder and the relatively low computational costs, which allow one to implement a system for real-time applications. Codec symmetry implies comparable computational loads at both the encoder and decoder. Because JPEG encodes each frame separately, i.e., only spatially, its performance is inferior to the standards that remove temporal redundancy between frames. MPEG-I is a video compression standard developed for consumer products and is targeted to operate at 1.5 Mbits/sec. This standard was designed to allow scanning in both forward and reverse directions through a video sequence, as in a VCR. MPEG-I removes source redundancy both temporally and spatially and the encoder/decoder constructs are highly un-symmetric, because of the coding tools that remove temporal redundancy. With the growth of the video communications industry, a new standard called MPEG-II has been designed to provide TV quality video at an average bit rate of 3 Mbits/sec and will support various service applications. Some applications may require VCR-like scanning functions, while others, such as those targeted to the cable industry, do not need this overhead scanning information added to their signals. The North American HDTV standard will be based on MPEG-II. The H.261 ($p \times 64$) standard was developed for use on ISDN networks and to operate on channels with bit rate capacities which are allocated in multiples of 64 kbits/sec. The standard uses the abbreviation " $p \times 64$ " to represent its variable transmission rate of

¹Joint Photographic Expert Group

²Moving Pictures Experts Group

integer multiples of 64 kbits/sec. This rate was chosen because telephony standards often segment channels into bit rates of 64 kbits/sec. The performance of the system is tied to the integer value of p ; the larger the value of p , the higher the bit rate and quality of the decoded video. The largest value of p in the standard is 32. For a short discussion on integrated circuits that implement these standards, see Appendix A.

The standards mentioned above use many techniques to achieve data compression. The amount of compression varies depending on the system and the video sequence. This is due to the use of different coding tools and the type of video a standard is targeted to service, because video statistics can change drastically from one sequence to the next. The “tool-kit” of methods used in video compression is shared by other signal processing tasks such as speech coding. A partial list of these tools includes: pulse code modulation (PCM); differential pulse code modulation (DPCM); scalar quantization and vector quantization (Gersho and Gray 1992); block- and pixel-based motion compensation (Walker and Rao 1984); the discrete cosine transform (DCT) (Clarke 1985); the Karhunen-Loève transform (KLT) (Clarke 1985); subband filtering coders (Vetterli 1984; Woods and O’Neil 1986); pyramidal coders (Burt and Adelson 1983); quadtree coders; interpolation techniques; entropy coders; and run-length coders. Many of these coding tools will be described in this thesis.

Video coders can be divided into lossless and lossy classes. In a lossless mode, the source information is not altered in the coding process; an application requiring very high quality and/or lossless coders is that used by the medical profession for imaging. In a lossy coder, distortion may be added to gain compression. Sometimes, the added distortion is masked by aspects of the video sequence information not perceived by the human visual system (HVS). Other than for archival purposes, codecs used for entertainment purposes, such as TV, usually employ lossy coding schemes.

Lossless and lossy codecs can be further classified by a fixed or variable bit rate transmission mode. In a fixed rate system, each data symbol is represented by a binary code of fixed length; in a variable rate system this is not required. Each system has its advantages and disadvantages. Fixed rate coders are much easier to resynchronize, especially when channel errors occur. Variable rate codecs generally achieve higher compression ratios than those using fixed rates, but buffering a variable rate system to a fixed rate channel requires feedback to the encoder so as not to overflow the

buffer and lose data. A fixed rate coder is often used when high error protection is required or when probability density function optimized quantizers are implemented. Variable rate systems most often result when entropy coding tools have been used.

There is a basic problem with current state of the art video codecs. For example, consider the well studied and commonly used MC-DCT codec configuration. This system is similar to the MPEG standards, which use block based motion compensation (MC), the DCT, interpolation, scalar quantization, and entropy coders. The operations required to encode a frame of video for the block based MC-DCT codec are as follows:

1. the encoded frame is segmented into equal sized blocks,
2. a distortion measure search of all possible matches in a windowed area of the previously encoded frame is performed,
3. the search location with the lowest distortion is selected,
4. the spatial displacement of the best match from the current block's location is recorded and defined as the block's motion vector,
5. the pixel by pixel difference between each block and its best match block from the previous frame is used to construct a new frame called the displaced frame difference (DFD),
6. the DCT is performed on each DFD block resulting in a set of transform coefficients,
7. the coefficients in every block are quantized,
8. and entropy coding is performed.

A major disadvantage of this system configuration is that characteristic visual impairments, called blocking effects, result from coarse quantization of the DCT coefficients. These effects can be perceived during periods of high motion or when a coarse quantizer is selected by the feedback loop from a fixed-rate channel buffer. This poor performance might cause one to question whether or not it is possible to use other methods to code the MC displaced frame difference (DFD) images in order to reduce

or remove the blocking effects with the same or better codec performance as DCT based techniques. One idea is to use methods that spread the quantization noise about a region of the frame and away from the edges of the MC blocks. Research measuring the performance of MC with alternative coding tools, other than the DCT, is needed. For example, there is a lack of results showing the performance of systems that integrate MC with subband filtering and pyramidal coders.

Another important property of video codes is “multiresolution”. In these systems, the signal is split into a tiered quality system, where an increase in quality results from the use of increased information from the encoder. Systems that perform frequency decompositions are conducive to such decompositions, because each frequency grouping can be considered a part of the multiresolution signal. If we divide video into frequency bands, we find that most of the information tends to be in the low frequency bands. If we let the lowest frequency band be our base signal in a multiresolution video service, the addition of each higher frequency band increases the resolution. The MC-DCT codec combination can be decomposed into a multiresolution video signal, but the lower resolution versions have severe blocking noise. Alternatively, subband filtering and pyramidal codecs can readily be used in multiresolution service applications.

If we consider subband decompositions, Karlsson and Vetterli (1988a) have shown that three-dimensional subband filtering performs poorer than MC by itself, but they do not discuss how a hybrid system would perform using both coding tools. Paek, Kim, and Lee (1992) recently investigated the integration of MC with spatial subband filtering. Their results showed that subband filtering followed by MC on the lowest subband was better than MC followed by subband filtering. PCM was used on the high frequency subbands. Their results, however, are inconclusive because they studied only one rate. Despite these results, the question still remains as to how a three-dimensional subband decomposition performs with MC and in what order it is best to integrate the two coding tools. The integration of these two tools is of interest because such a system would not be subject to blocking effects noise and a multiresolution signal could be constructed.

This thesis describes the results of an empirical investigation into the integration of motion compensation with subband filtering. The purpose of this work is to provide a guide on how to integrate MC with temporal and spatial subband filtering. This thesis

uses existing signal processing theory where possible but, because no good model of video information exists, an experimental investigation is justified.

This work is partitioned into five sections. First, a background of video coding tools is described in detail in Chapter 2. Second, performance measures and statistical measures are defined in Chapter 3. Third, a detailed description of each designed codec is given in Chapter 4. Fourth, simulation results are reported and discussed in Chapter 5. Finally, conclusions and a discussion with suggestions for further work are presented in Chapter 6.

Chapter 2

Background of Video Coding Tools

A background study of several design tools for video source codecs is presented in this chapter. Included is a discussion of sampling, quantization, PCM, DPCM, motion compensation, transform coders, run-length coders, entropy coders, subband filtering, and pyramidal coders. Most of these design tools are used in the codecs studied in this thesis. To reiterate, the purpose of a video source coder is to reduce the transmitted data rate by exploiting the redundancy in both the spatial and temporal domains of a video sequence.

The coding methods described below can be classified as either inter-frame or intra-frame, encoding data either between or within frames respectively. Motion compensation and temporal subband coding methods are inter-frame techniques, whereas many transform coders and two-dimensional subband coders are in the intra-frame class. Most video coding methods use both inter-frame and intra-frame coding schemes to remove both temporal and spatial redundancies.

2.1 Sampling

This thesis deals with digital video compression, which necessarily operates on time-discrete data; video information perceived in the natural world is, however, both time-continuous and amplitude-continuous. The process of transforming continuous time information to discrete time is called sampling and is achieved by using video cameras and pre-encoder video-processors to take samples of the video information at specified intervals of time. For video, a field (frame) of two-dimensional data is

recorded each successive sample period and, when the fields are put in sequential order, they represent the video scene information. This is in contrast to sampling of speech or audio information, where only one data point is taken each sample period, i.e., the amplitude of the sound wave at the sample time.

The rule that specifies the sampling rate so that the sampled data can be used to reconstruct the time-continuous information without distortion, noticeably the effects of aliasing distortion, is called the Nyquist sampling theorem. This theorem states that the sampling rate must be twice the highest frequency found in the source if aliasing is to be avoided. The temporal sampling rate of film-based movies is 24 frames per second, but this rate is several times below the Nyquist sampling rate and the effects of aliasing are noticeable. One example of aliasing can be seen when a forward-moving vehicle's wheels appear to move backwards. In addition to temporal sampling, each data field (frame) of video must be sampled spatially. This sampling is also subject to aliasing when the spatial sampling rate falls below the Nyquist rate. The effects of spatial aliasing are much smaller than those of temporal aliasing because the iris and lens used in video cameras acts as low-pass filters, which removes most spatial frequencies above the spatial Nyquist rate.

Many different three-dimensional sampling patterns have developed (Dubois 1985). Some take advantage of the characteristics of the human visual system (HVS). The most basic sampling mode is called progressive. In this mode, a complete frame of data is collected each sample period, so that the video sequence is represented by a sequence of complete still frames. This type of sampling is used for movie theater films. An alternate sampling mode, called interlaced, is used in the National Television System Commission (NTSC) television standard (Netravali and Haskell 1988). An interlaced sequence of frames is one where alternative scan lines in the picture are present in each successive frame. For example, the odd frames would contain the odd scan lines and, the even frames, the even scan lines. For the same overall data rate, interlaced sampling thus has twice the temporal sample rate as progressive sampling. The NTSC standard has a temporal sampling rate of 60 fields/sec. Because of frame flicker masking effects in the HVS, interlaced sampling has an advantage over progressive sampling when the sampling rate is low.

Video can be either monochrome or color. For monochrome video, each sample pixel amplitude represents gray scale intensities between black and white. However,

for color video, the pixels are usually represented by three sample amplitudes, each representing a primary, or near primary, color. As a result, the data rate increases for color, but is usually not raised by a factor of three. Instead, using color transforms that reduce the required bandwidth (Netravali and Haskell 1988), the color codec transmission data rate increases only to about 1.5 times that of a monochrome codec.

The source coders used in this work assume that the frames are sampled progressively and that the pixels in the spatial field are monochrome. In addition to sampling, the amplitude-continuous values in the spatial field must be represented by numbers with finite precision. The process that does this is described next.

2.2 Quantization

Once the video camera has sampled the video scene, how are the floating-point pixel values in each frame represented? The process that transforms amplitude-continuous to an amplitude-discrete format is called quantization (Jayant and Noll 1984). This process takes an infinite, or high, precision number and represents it with a finite set of discrete numbers. This process is a nonlinear and an information lossy operator. Because of this, the quantizer greatly affects both the overall distortion and the data transmission rate. One usually seeks to minimize distortion and quantizer design is therefore concerned with an optimization process that finds the lowest distortion between the original and quantized data for a given rate (or vice-versa).

Approaches to quantization can be divided into two different groups called scalar (Jayant and Noll 1984) and vector (Gersho and Gray 1992) quantization. In scalar quantization, each sample in the data is quantized separately as a unit on its own, whereas, in vector quantization, vectors of samples are quantized together. The design and implementation of a vector quantizer is more complex than that of a scalar quantizer, but the performance and benefits of using vector quantizers are significant. One major benefit of a vector quantizer is that samples can be quantized at non-integer bit rates. Scalar quantizers were used exclusively in this work, however, since they are adequate to compare the performance of the coding methods under test.

The basic structure of a scalar quantizer is to make a many-to-one mapping of each input real number x to a finite set of output real numbers $y_k : k = 1, 2, \dots, L$, where $L = 2^m$ and m represents the bit rate. Figure 2.1 shows a block diagram of

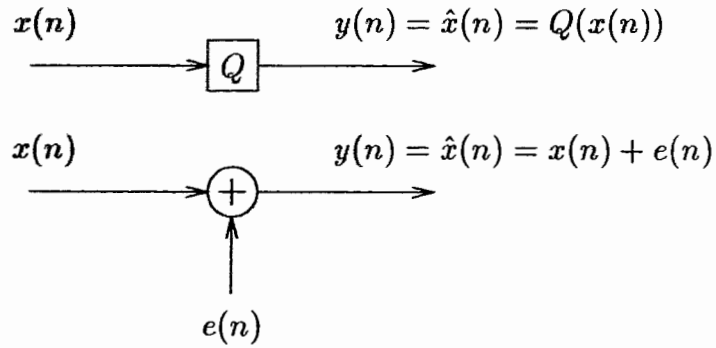


Figure 2.1: Quantizer Block Diagram and Noise Model

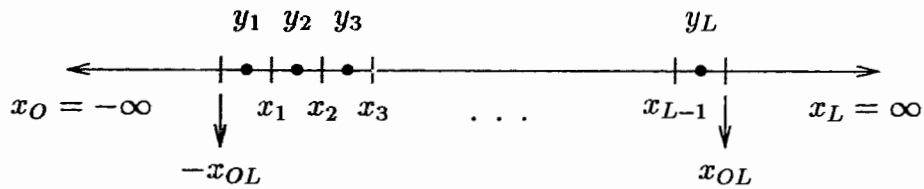


Figure 2.2: Many-to-one Scalar Quantizer Mapping

the process, where $Q(\cdot)$ denotes the mapping from $x(n)$ to $y(n)$ and n represents the time index. The bottom half of Figure 2.1 shows how the distortion can be viewed as a linear combination of the input signal $x(n)$ and an error signal $e(n)$. The mapping function $Q(\cdot)$ can be more precisely defined as

$$Q(x) = y_k \quad \text{iff} \quad x \in [x_{k-1}, x_k) \quad , \quad (2.1)$$

where the points $x_k : k = 0, 1, \dots, L$ represent decision boundaries between output y_k values. Figure 2.2 shows this mapping on the real line. In addition, the magnitude of the nonlinear quantization error is

$$e = |y_k - x| \quad . \quad (2.2)$$

Approaches to choosing the output values and decision boundaries are varied and numerous. Most design methods (Jayant and Noll 1984) try to exploit the statistics of the source, such as the variance and its probability density function (pdf), to reduce the reconstructed distortion. These methods assume that the source is either a stationary or a wide-sense-stationary process. For non-stationary sources, adaptive

quantizers have been developed (Jayant and Noll 1984). Common types of scalar quantizer strategies are termed uniform, non-uniform, and logarithmic (logarithmic is a special case of non-uniform strategies).

In uniform quantization, the intervals between output values are made equal to a constant value, called the step-size (Δ), and the decision boundaries are set to the mid-points between reconstructed values. The performance of a uniform quantizer is subject to both granular and overload distortion. Referring back to Figure 2.2, granular distortion is the distortion measured when input values fall between $x \in [x_{-OL}, x_{OL}]$ and overload distortion is measured when the input falls outside this interval. It can be shown that the quantization noise variance is $\sigma_e^2 = \frac{\Delta^2}{12}$ and is independent of the input variance using the following assumptions (Oppenheim and Schaffer 1989):

1. the error sequence $e(n)$ is a sample sequence of a stationary random process,
2. the error sequence is uncorrelated with the sequence $x(n)$,
3. the random variables of the error process are uncorrelated, i.e., the error is a white-noise process,
4. the probability distribution of the error process is uniform over the range of quantization error,
5. and the source does not exceed the quantizer range.

If the pdf is used to optimize the quantizer, then the quantization noise is strongly dependent on the input variance.

In non-uniform quantizers, the intervals between output values need not be a constant. The design of non-uniform quantizers is often achieved via an iterative procedure, such as the Lloyd-Max algorithm (Gersho and Gray 1992) and, for a given number of output levels, it is possible to obtain a quantizer with a smaller reconstructed error variance than the uniform quantizer error performance. This is because these quantizers allocate more output values where the probability of occurrence is high and vice-versa for low probability of occurrence.

In logarithmic quantization, the input source is companded and then uniform quantized. The companding operation uses a law that tends to allocate more output levels to low amplitude input values, and fewer output levels to high amplitude input

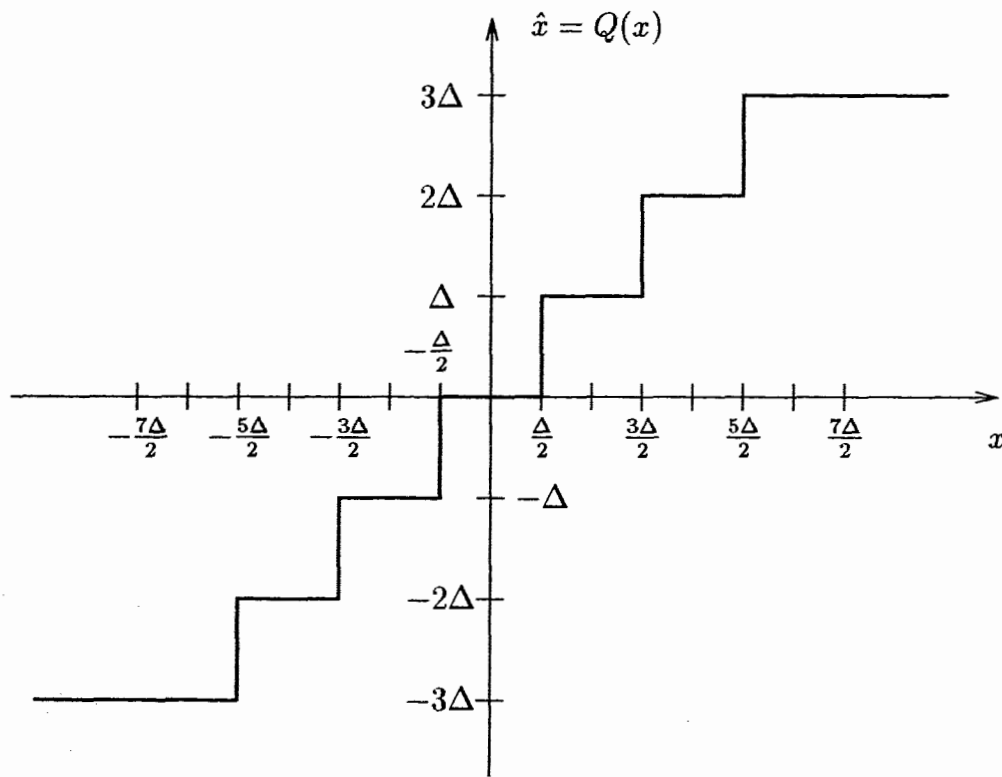


Figure 2.3: Midtread Seven Level Uniform Quantizer Transfer Function

values. The result is to flatten the input pdf, which results in an improved dynamic range. These quantizers are used for coding sources, such as speech, where the input variance is not known in advance and may change over time. The benefits of logarithmic quantizers are their insensitivity to changes in the input variance.

Even though the performance of a uniform quantizer may be lower than a non-uniform quantizer, its performance when cascaded with an entropy coder is asymptotically as good as, or better than, that obtained with other methods (Jayant and Noll 1984; Gersho and Gray 1992). Entropy coders are discussed later in this chapter.

A further quantizer characteristic is that of either midtread or midrise. Midtread quantizers have an output value at zero, whereas midrise quantizers have a decision boundary at zero. For sources with a zero mean symmetric pdf that are peaked at zero, the performance of a midtread uniform quantizer is generally better than that of a midrise quantizer, because of the midtread output value at zero. Figure 2.3 shows a midtread seven level uniform quantizer transfer function. Another option to quantizer designs includes the implementation of a dead zone in the quantizer. This method is

sometimes added when the source pdf is highly peaked at zero and when most of the values about zero can be considered noise.

Sometimes quantizers are implemented inside feedback loops. Such implementations are designed using simplifying assumptions, because the quantizer nonlinearities are difficult to describe in analytic design equations.

2.3 PCM and DPCM

Two fundamental coding methods are pulse code modulation (PCM) and differential pulse code modulation (DPCM) (Jayant and Noll 1984). PCM is the simpler of the two and consists of a source sampler followed by an amplitude quantizer. DPCM is more complex than PCM, but generally has a higher coding gain due to the predictive coding methods that remove redundancy/correlation in the source data stream. Because both sampling and quantization have been defined in the previous sections, no further discussion will be given to PCM.

Differential PCM is a coding scheme that transmits a signal consisting of the quantized differential between the input and a prediction of the input. The complexity of a DPCM codec is a function of the predictor algorithm. Low complexity predictors tend to be time-invariant, whereas, high complexity predictors can use adaptive time-variant predictors. The basic block diagram of a DPCM coder/decoder is shown in Figure 2.4. In this figure, $x(n)$ and $y(n)$ denote the input and reconstructed output discrete time signals. The difference signal $d(n)$ is the input to the quantizer and is defined as

$$d(n) = x(n) - \hat{x}(n) \quad . \quad (2.3)$$

The difference signal is then quantized to form $u(n)$, which is then transmitted via the channel to the decoder. Remembering the quantizer model in Figure 2.1, $u(n)$ is defined as

$$u(n) = d(n) + e(n) \quad . \quad (2.4)$$

The quantizer is located within the predictor feedback loop so both the encoder and the decoder make their predictions based on the same signal, $y(n)$ (assuming there are no channel transmission errors). Finally, the reconstructed output signal is defined as

$$y(n) = u(n) + \hat{x}(n) \quad , \quad (2.5)$$

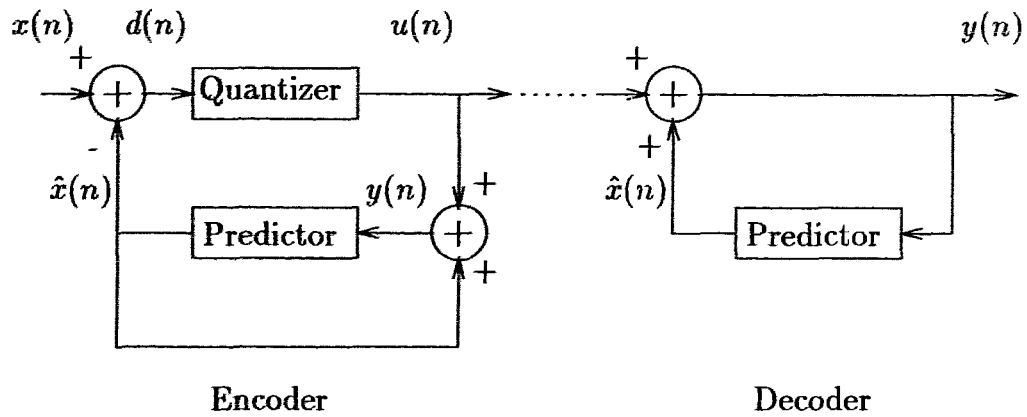


Figure 2.4: Block Diagram of a DPCM Coder and Decoder

which is the quantized difference signal plus the predicted sample value. Much of the correlation in the input signal is removed by the predictor and, as a result, DPCM can be thought of as a whitening process. The transmitted quantized difference signal also tends to have a lower variance than the input signal, which generally implies a coding gain over PCM.

A common application of DPCM in video compression is the coding of spatial frames. In the two-dimensional case, the predictor design must use previously encoded pixels in the neighborhood of the current pixel to calculate its prediction value. Much of the complexity lies in the predictor's design, especially if an adaptive predictor is used. Predictors can be classified as linear or nonlinear. A good introduction to prediction theory is described both by Gersho and Gray (1992) and Jayant and Noll (1984). Gersho and Gray define an optimal affine predictor, which has good performance when predicting sources with a nonzero mean, such as those for images. This type of predictor was used in this work and is defined below.

Suppose we are given an N -dimensional vector $\mathbf{X} = (X_{n-1}, X_{n-2}, \dots, X_{n-N})^T$ and wish to predict a K -dimensional vector $\mathbf{Y} = (Y_n, Y_{n+1}, \dots, Y_{n+K-1})^T$, where n is the time or spatial step index. Then, an optimal affine predictor is defined as

$$\hat{\mathbf{Y}}(\mathbf{X}) = \mathbf{A}\mathbf{X} + \mathbf{b} \quad , \quad (2.6)$$

where \mathbf{A} represents the matrix of predictor coefficients and \mathbf{b} represents a vector of constant terms. For ease of development, let the matrix \mathbf{A} be defined as

$$\mathbf{A} = [\bar{\mathbf{a}}_1, \bar{\mathbf{a}}_2, \dots, \bar{\mathbf{a}}_K]^T \quad , \quad (2.7)$$

$x_{i-1,j-1}$	$x_{i,j-1}$
$x_{i-1,j}$	$x_{i,j}$

Figure 2.5: Spatial Prediction Pixel Orientation

where

$$\mathbf{a}_k = (a_{k_1}, a_{k_2}, \dots, a_{k_N})^T, \quad (2.8)$$

and let the vector \mathbf{b} be defined as

$$\mathbf{b} = (b_1, b_2, \dots, b_K)^T. \quad (2.9)$$

Any solution \mathbf{A} of the equation

$$\mathbf{K}_X \mathbf{A} = E[(\mathbf{Y} - E(\mathbf{Y}))(\mathbf{X} - E(\mathbf{X}))^T] \quad (2.10)$$

will provide a set of optimal predictor coefficients (in the mean squared sense) to the system $\mathbf{Y} = \mathbf{A}\mathbf{X} + \mathbf{b}$, where the covariance matrix \mathbf{K}_X is defined by

$$\mathbf{K}_X = E[(\mathbf{X} - E(\mathbf{X}))(\mathbf{X} - E(\mathbf{X}))^T] \quad (2.11)$$

and

$$\mathbf{b} = E[\mathbf{Y}] - \mathbf{A}E[\mathbf{X}]. \quad (2.12)$$

If \mathbf{K}_X is invertible, then

$$\mathbf{A} = E[(\mathbf{Y} - E(\mathbf{Y}))(\mathbf{X} - E(\mathbf{X}))^T] \mathbf{K}_X^{-1}. \quad (2.13)$$

For cases when \mathbf{K}_X is almost singular, a mathematical method such as the Singular-Value-Decomposition (Press et al. 1988) must be used for finding the solution.

Using the above definitions, a spatial one-step three-tap affine predictor with input samples taken from the spatial orientation shown in figure 2.5 is defined. The predicted vector $\hat{\mathbf{Y}}$ becomes a scalar and is defined as

$$\hat{\mathbf{Y}} = (\hat{Y}_1) = \mathbf{a}_1^T \mathbf{X} + b. \quad (2.14)$$

Transforming the affine predictor notation back to the DPCM notation used in Figures 2.4 and 2.5, the estimate \hat{x} of x becomes

$$\hat{x}_{i,j} = a_{11}x_{i-1,j} + a_{12}x_{i-1,j-1} + a_{13}x_{i,j-1} + b. \quad (2.15)$$

This predictor was used for regions of the image below the top row and right of the left most column of pixels in the image. For the remaining regions, the following one-step one-tap linear predictors are defined:

$$\begin{aligned}\hat{x}_{i,1} &= a_h x_{i-1,1} + b_h \\ \hat{x}_{1,j} &= a_v x_{1,j} + b_v \\ \hat{x}_{1,1} &= 127\end{aligned}\tag{2.16}$$

The next section describes motion compensation. This technique can be considered a specially modified case of temporal DPCM, where pixel blocks of size $K \times K$ are predicted with pixel blocks of size $K \times K$ from a varied region in the previous frame, and the predictor taps are always “1”, i.e., $A = I$.

2.4 Motion Compensation

Motion compensation (MC) is an important component of modern video codecs such as MPEG (LeGall 1991) and H.261 (Liou 1991). Numerous MC methods exist; some use feature-based models and others use the idea of optic-flow (Aggarwal and Nandhakumar 1988). These two methods use the two-dimensional image sequence characteristics to create three-dimensional motion information, use extremely complex algorithms, and are computationally expensive. Practical MC methods, which can be implemented in real time, generally operate at the pixel level. Two techniques are called block and pixel MC (Netravali and Haskell 1988). MC on a block rather than a pixel level is less accurate but, depending on the block size, can be significantly less computationally expensive. Pixel MC can describe the motion to higher accuracy than block methods, although block methods can still estimate most gross motion. Block MC methods may require extra bits when there is a bad prediction, but this is more than offset by the improved efficiency; block MC methods are therefore most often used in coding applications.

The basic approach taken by block MC is to partition the current frame into $l \times l$ blocks. For each block (refer to Figure 2.6), a search is performed to find the $l \times l$ region in the previous frame that provides the best match. Then, instead of processing the original frame, the coder transmits the differences between the matched blocks — called the displaced frame difference (DFD) signal — as well as vector information

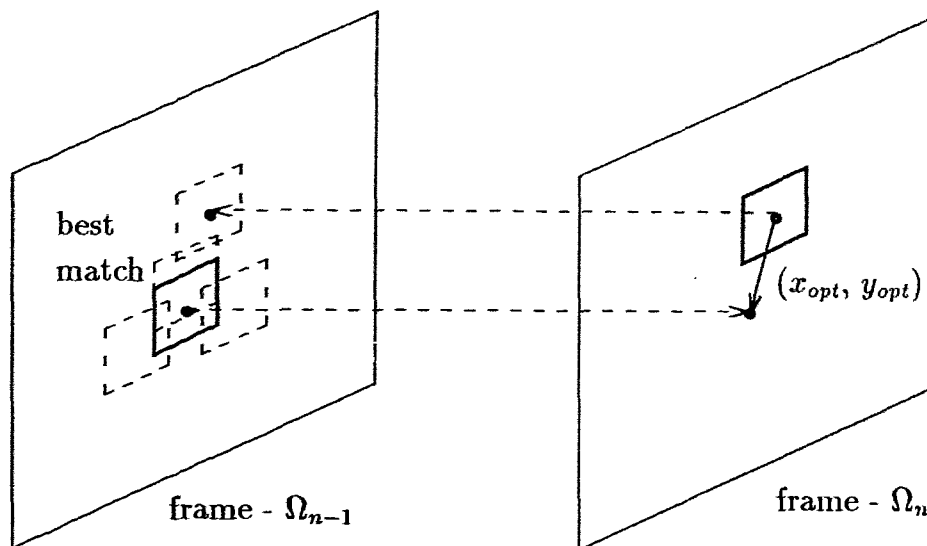


Figure 2.6: Block Motion Compensation

describing the location of the best matched block in the previous frame. Although this technique substantially improves the codec performance, the cost in terms of search time can be prohibitive due to the computational load. As a result, numerous suboptimal search strategies have been developed, i.e., searching only a small window of the previous frame, and the three step search method (Koga et al. 1981).

Generally, all search methods, including what is called the full-search method, search only for matches from a windowed area in the previous frame. The window is usually centered about the block in the frame currently being encoded. When using windowed methods, the search window size used is an extremely important parameter, since it affects both the quality of the match and the computational burden of the search. As the window area grows from zero, the performance increases up to a point and then saturates. Video coders typically have square windowed search regions that span ± 15 or ± 31 pixels in the previous frame.

A mathematically rigorous definition of the windowed full-search method is as follows. Let Ω represent the sequence of frames to be encoded, and Ω_n frame n . Then, let the spatial horizontal and vertical frame dimensions be W and H pixels respectively. Furthermore, let a macro-block consist of an $N \times M$ rectangular matrix of pixels, where N and M represent the horizontal and vertical block size dimensions respectively. In most MC systems, $N = M$. Therefore, without loss of

information, we will assume block sizes of $N \times N$ pixels, which implies N^2 pixels per block. Now, when encoding frame n , the frame Ω_n is partitioned into a set of non-overlapping macro-blocks denoted by $\Omega_n(\mathbf{r}_i)$, where $\mathbf{r}_i : i = 1, 2, \dots, \frac{WH}{NM}$ is a vector describing the Cartesian coordinates of the upper left pixel in each macro block. Next, let the windowed search area \mathcal{W}_i for each block be a rectangular region centered about the point \mathbf{r}_i spanning horizontally $\pm p$ and vertically $\pm q$ pixels and located in Ω_{n-1} . If $N = M$, the search size variables p and q are, in general, equal. A square search region of $(2p + 1) \times (2p + 1)$ pixels will be assumed for the remainder of this work. Let \mathbf{mv}_i represent a vector describing the spatial displacement (i.e., motion) of the $\Omega_{n-1}(\cdot)$ that is referenced relative to position \mathbf{r}_i in the search window \mathcal{W}_i . Now, the MC algorithm finds the

$$\operatorname{argmin}_{\mathbf{mv}_i} \left(\sum_{\mathbf{mv}_i \in \mathcal{W}_i} d(\Omega_n(\mathbf{r}_i) - \Omega_{n-1}(\mathbf{r}_i - \mathbf{mv}_i)) \right) \quad (2.17)$$

according to the matching criterion $d(\cdot)$. In most cases, $d(\cdot)$ represents the l_1 or l_2 norms calculated using all the pixels in the two blocks $\Omega_n(\mathbf{r}_i)$ and $\Omega_{n-1}(\mathbf{r}_i - \mathbf{mv}_i)$. In the literature, the one and two norms are referred to as the absolute error

$$ABS(\mathbf{r}_i) = \frac{1}{MN} \sum_{x=1}^M \sum_{y=1}^N |\Omega_n(\mathbf{r}_i, x, y) - \Omega_{n-1}(\mathbf{r}_i - \mathbf{mv}_i, x, y)| \quad (2.18)$$

and the mean squared error

$$MSE(\vec{r}) = \frac{1}{MN} \sum_{x=1}^M \sum_{y=1}^N (\Omega_n(\mathbf{r}_i, x, y) - \Omega_{n-1}(\mathbf{r}_i - \mathbf{mv}_i, x, y))^2 \quad (2.19)$$

respectively, where $\Omega_n(\mathbf{r}_i, x, y)$ represents the pixel amplitude of pixel (x, y) in the macro-block i and in frame n . Given the best motion vector \mathbf{mv}_i for block i , the i th DFD block is defined to be

$$\Omega_{DFD_n}(\mathbf{r}_i) = \Omega_n(\mathbf{r}_i) - \Omega_{n-1}(\mathbf{r}_i - \mathbf{mv}_i) \quad (2.20)$$

Finally, the encoder transmits the quantized Ω_{DFD_n} and the motion vector for each block.

Enhancements and modifications to the full-search exist. In conditional replenishment coding, DFD blocks whose energy is below a given threshold are not transmitted; and/or if the energy in the original block has lower energy than its corresponding

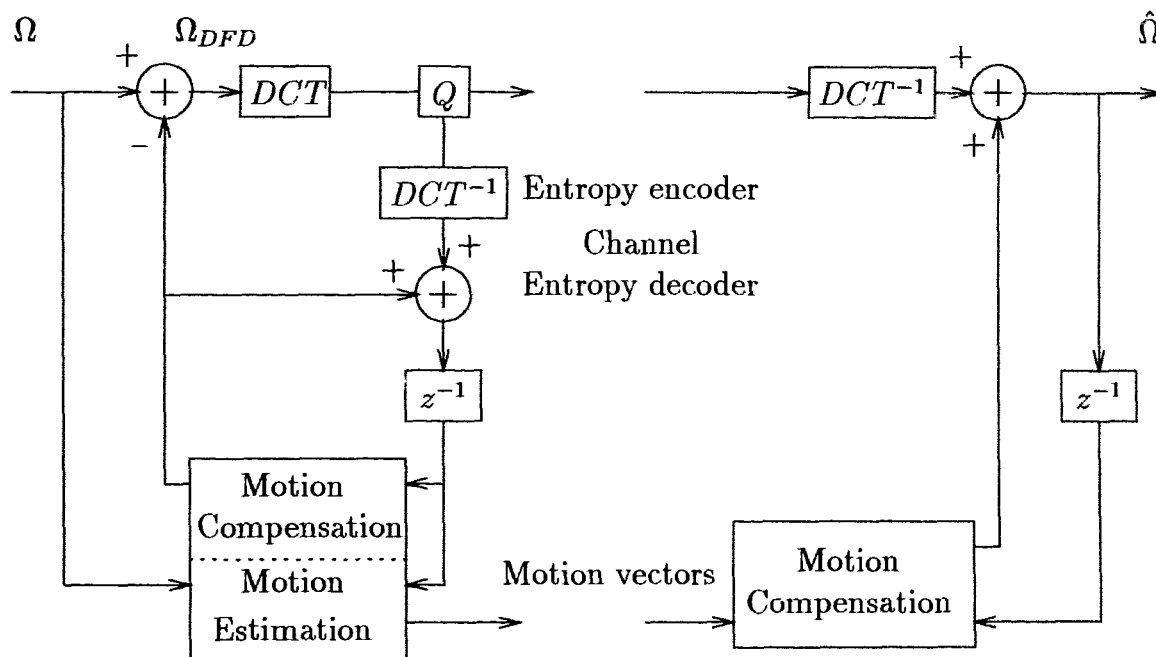


Figure 2.7: A Motion Compensation Codec

DFD block, the original block is transmitted instead. Alternative search methods are numerous, and two examples of these are the three-step (Koga et al. 1981) and decimated (Zaccarin and Liu 1992) searches. These methods reduce the number of block compares at the cost of no longer finding the optimal motion vector. In addition to searching the previous frame for block matches, Gothe and Vaisey (1993) showed that energy in the DFD frame can be reduced by searching more than one previous temporal frame with a reduction in the total number of block compares.

Once an MC method has calculated the DFD, additional coding tools are used to remove additional redundancy from the DFD frame. The use of the discrete cosine transform, or a vector quantizer followed by entropy coders, are possible choices for further DFD frame coding. Figure 2.7 shows a block diagram of such a codec. Because quantizers are used, both the encoder and decoder use a previously reconstructed frame as the searched frame in order to prevent the propagation of quantization noise. To remove the propagation of distortion due to channel error in the MC decoder, frames not encoded with MC are periodically transmitted.

2.5 Transform Coding

Transform coders are important and are integral parts of modern video codecs such as MPEG (LeGall 1991) and H.261 (Liou 1991). Given an N -dimensional data set, a transform coder is used to alter the space so that the signal energy is compacted into as few components as possible. Many transform coders exist. The transform that minimizes the overall distortion for a given number of coefficients transmitted is the Karhunen-Loève transform (KLT) (Clarke 1985); however, it is not practical to use the KLT, because the transform basis vectors are dependent on the covariance matrix of the image data and must be recomputed each frame or every several frames (because of changing frame statistics). The most commonly used transform is the discrete cosine transform (DCT). This transform provides a significant reduction in computational complexity over the KLT, even though its performance is only slightly less than the KLT for autoregressive process with high correlation. The two-dimensional DCT is defined as follows:

$$F(u, v) = \frac{4C(u)C(v)}{n^2} \sum_{j=0}^{n-1} \sum_{k=0}^{n-1} f(j, k) \cos \left[\frac{(2j+1)u\pi}{2n} \right] \cos \left[\frac{(2k+1)v\pi}{2n} \right] \quad (2.21)$$

$$f(u, v) = \sum_{u=0}^{n-1} \sum_{v=0}^{n-1} C(u)C(v)F(u, v) \cos \left[\frac{(2j+1)u\pi}{2n} \right] \cos \left[\frac{(2k+1)v\pi}{2n} \right], \quad (2.22)$$

where

$$C(w) = \begin{cases} \frac{1}{\sqrt{2}}, & \text{for } w = 0 \\ 1, & \text{for } w = 1, 2, \dots, n-1 \end{cases} \quad (2.23)$$

Other non-optimal transforms include the discrete Fourier transform (DFT), Walsh-Hadamard transform (WHT), and the discrete sine transform (DST). The DFT is inferior to the DCT in that it performs worse with greater computational complexity than the DCT. The DFT requires the use of complex numbers, while the DCT uses only real numbers. The WHT performance is far from optimal, but it is simple to implement (Rabbani and Jones 1991). The DST has poorer performance than the DCT and, therefore, there is little reason for its use (Clarke 1985).

The use of the transform coders in video codecs is popular, especially since the DCT is used as an integral part of existing and proposed video standards. One of the reasons for the popularity of the DCT is that hardware integrated circuits that benefit from economies of scale have been developed. Despite these economic benefits, the

biggest disadvantage of using the DCT and other block transform coding tools is the blocking effects that appear in the reconstructed images due to quantization, most notably at low bit rates. Because of this and because it is hard to exploit the HVS, alternative video codecs that don't use transform coding need to be studied.

2.6 Run-length and Entropy Coding

Often, especially after quantization, further data compression is achievable via run-length and/or entropy coders. These methods work with the discrete symbols output from the quantizer and exploit the probability density function (pdf) of the data sequence.

The output of a quantizer is a sequence of indices that is often encoded simply by assigning a binary code to each index $k_j : j = 1, 2, \dots, L$. This implies a bit rate of $R = \log_2 L$ bits per pixel if L is a power of two, else R is rounded to the nearest integer larger than $\log_2 L$. This average bit rate usually decreases when run-length or entropy coding methods are applied. The source entropy is a measure that specifies the minimum bits needed to represent a source (under certain conditions, i.e., the rate is minimum if the source autocorrelation function is an impulse at zero); it is defined as

$$H = - \sum_{j=1}^L p_j \log_2 p_j \quad , \quad (2.24)$$

where p_j represents the probability of symbol k_j occurring in the source. This is a zeroth order measure, because it considers each symbol separately. Higher order entropy measures give lower average bit rates if the symbols in the source are correlated.

Run-length coding is a higher order encoding method that is often used if the data source has significant run-lengths of a particular data value, say zero. In this coding scheme, instead of sending n -zeros, one would send the number of consecutive zeros, then send the number of consecutive non-zero values followed by the respective non-zero data values. This procedure is repeated for the remainder of the data stream. Additional symbols required to describe the run-lengths are introduced to form a new source. Generally, one either assigns binary codes to these new symbols or else further encodes the new source with an entropy coder. The former technique is used in the JPEG standard.

In summary, entropy coding methods approach the source entropy by using variable length source coding techniques. These techniques use *a priori* knowledge of the input symbol's relative frequency to code the data. In other words, symbols with high relative frequency in the data stream are coded with short code words and the converse for symbols with low relative frequency. Because the resulting code is variable length, good error protection is needed, otherwise a single bit error results in errors in the rest of the data stream. One common entropy coding method is Huffman coding (Huffman 1952). In this technique, a simple algorithm that uses the relative frequency of the data symbols is used to create a Huffman table, which is a mapping between the symbol indices and calculated binary codes. Once the table is constructed, encoding is achieved via a lookup table.

2.7 Subband and Pyramidal Coding

Two video coding methods that are not part of the existing compression standards are subband filtering (Woods and O'Neil 1986) and pyramidal coding (Wang and Goldberg 1989). These methods perform frequency decompositions of the video source and are thus conducive to being used in multiresolution coding implementations. These methods are not subject to blocking noise effects as in transform coders such as the DCT; instead, they spread or smear the quantization noise among numerous pixels in the image, which is generally less perceivable by the HVS.

Subband filtering is an operator that decomposes an N -dimensional source into M frequency bands. One of the benefits of subband filtering video is the compaction of the signal energy into the lowest frequency subbands. This occurs because much of the video information is at low frequencies. Since most of the signal energy is in the lowest frequency bands, this information can be quantized with low distortion, whereas, coarser quantizers can be used for the bands that contain little information.

Subband filtering was first developed for frequency decompositions for speech coding, later applied to image compression by Woods and O'Neil (1986), and to video by Karlsson and Vetterli (1988a, 1988b). To describe a subband filtering system, first consider a one-dimensional signal source. Decomposition of this source into frequency bands is called analysis subband filtering. Once the source is filtered, further coding tools like quantization and entropy coders can be used. The reconstruction of

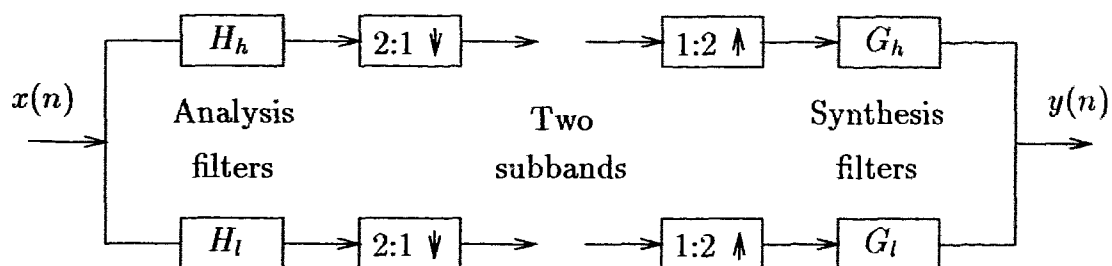


Figure 2.8: Subband Analysis/Synthesis Filter Structure

the signal source from its subband representation is called synthesis filtering. Figure 2.8 shows a block diagram of a two-band frequency decomposition subband filtering system, where the source is high/low pass filtered followed by down-sampling each subband by a factor of two. Because of the down-sampling, the resulting two signals have the same number of samples as the original. The down-sampled subbands are up sampled by two and synthesis filtered to reconstruct the original source. It is known that the decimation process results in aliasing, but if appropriate filters are used, the aliasing is canceled in the synthesis filtering, provided there is no quantization in the process.

Design procedures for generating two band filter sets that perform perfect reconstruction without aliasing errors are well documented in the literature. Johnston (1980) defined quadrature mirror filters (QMF), Smith and Barnwell (1986) defined conjugate quadrature filters (CQF), and LeGall and Tabatabai (1988) defined short kernel perfect reconstruction filters that can be designed to achieve at least perfect reconstruction. For example, consider the high/low pass filters shown in Figure 2.9. If these frequency responses represent the analysis filters H_l and H_h , each of the above design procedures above would specify the frequency response of the analysis filters G_l and G_h . Using the notation $h(n)$ and $H(z)$ to represent the filters impulse and Z-transforms respectively, and where N is the filter impulse length, the constraints

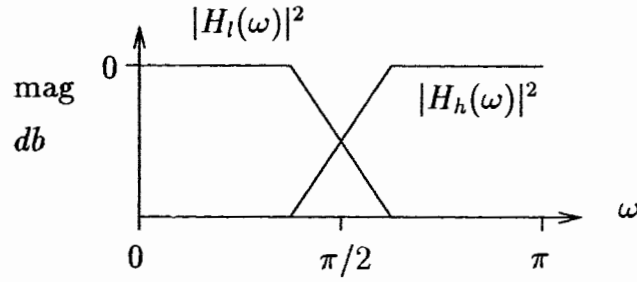


Figure 2.9: Analysis Filter Set Frequency Response

on a QMF are given by

$$\begin{aligned}
 h_l(n) &= h_l(N-1-n), \quad n = 0, 1, \dots, N/2-1 \\
 h_h(n) &= -h_h(N-1-n), \quad n = 0, 1, \dots, N/2-1 \\
 H_h(z) &= H_l(-z) \\
 G_l(z) &= H_h(-z) \\
 G_h(z) &= -H_l(-z)
 \end{aligned} \tag{2.25}$$

on a CQF are given by

$$\begin{aligned}
 h_h(n) &= (-1)^n h_l(N-n) \\
 g_l(n) &= h_l(N-n) \\
 g_h(n) &= -(-1)^n h_l(n)
 \end{aligned} \tag{2.26}$$

and on a perfect reconstruction filters are given by

$$\begin{aligned}
 G_l(z) &= H_h(-z) \\
 G_h(z) &= -H_l(-z)
 \end{aligned} \tag{2.27}$$

For all these filters, the reconstructed output sequence is delayed by an amount dependent on the filter impulse response length.

Subband filtering can be easily applied to higher dimensions, such as two- and three-dimensional image and video filtering respectively. For each added dimension, the appropriate filter sets must be designed. In general, N -dimensional filters can be used for the N -dimensional signal space, but the design and implementation of N -dimensional filters sets for subband filtering that reconstruct the original signal is very complex. These filters are called non-separable and have been studied by Vetterli (1984) and more recently by Bamberger and Smith (1992). Most practical subband

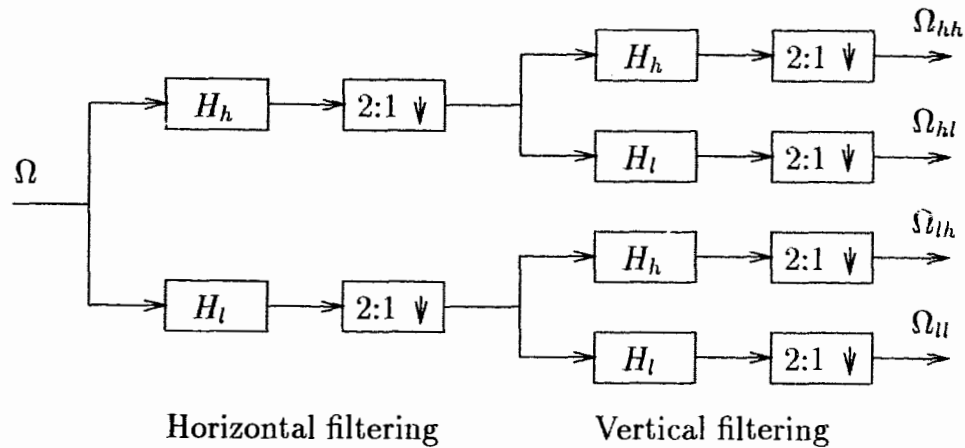


Figure 2.10: Spatial Analysis Subband Filtering

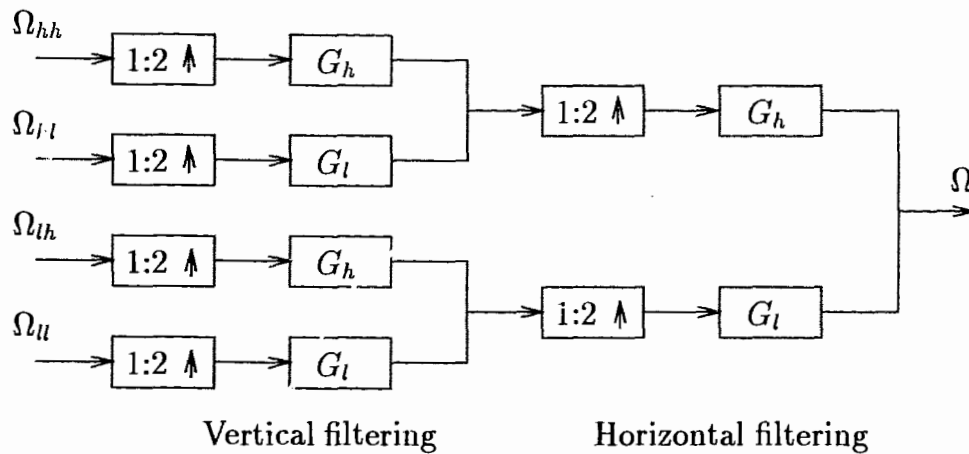


Figure 2.11: Spatial Synthesis Subband Filtering

filtering systems use separable one-dimensional filter sets that are easier to design and implement than non-separable filters. Because of this, only one-dimensional separable filters were used in this work. In the separable case, the signal space is filtered in each dimension to construct the subbands independently. The three filter types defined above produce separable filter sets. Figures 2.10 and 2.11 depict the filtering and down/up sampling steps required for spatial analysis and synthesis filtering of a video frame Ω respectively. Note: spatial subband filtering of finite sized images must take into account the implementation complexity of filter delays at image edges. One method is to support the filters by padding with zeros, but increasing the image size is undesirable. Therefore, a second method called circular filtering is commonly

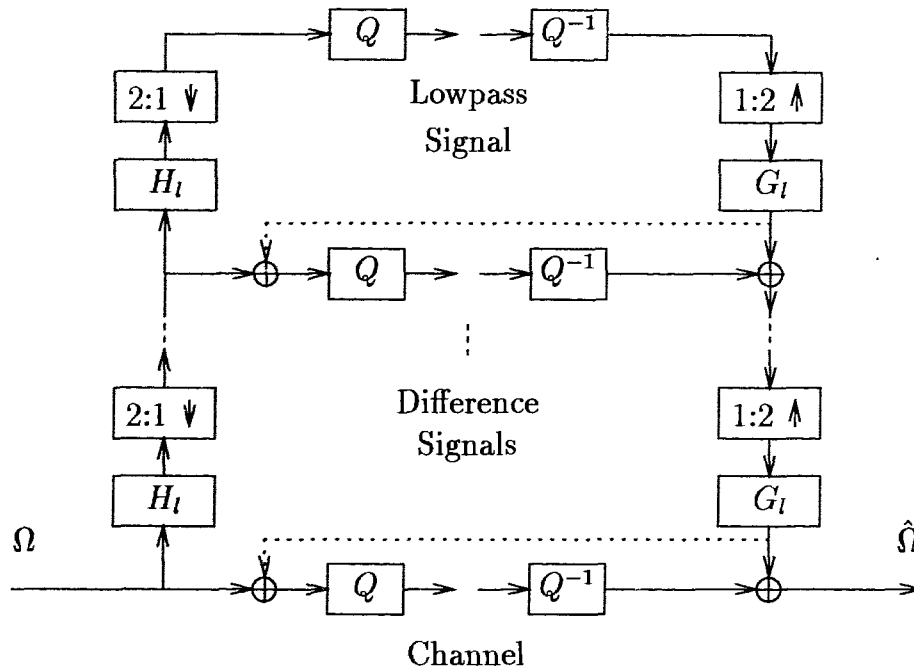


Figure 2.12: Block Diagram of a Pyramidal Codec

used (Smith and Eddins 1987). Circular filtering is the operation that connects the two ends of a horizontal row or vertical column of pixels together so filtering can be performed on the circular array of data without requiring padding. This method was used in this work for spatial filtering.

A second multi-dimensional coding method is based on pyramidal decompositions. First presented by Burt and Adelson (1983), spatial pyramidal coding can be used to transform a video frame or image into a decimated lowpass image and a number of difference images. Two advantages this system has over subband filtering are the use of feedback methods designed to reduce quantization noise and fewer constraints on the filters. Figure 2.12 show the basic block diagram of a pyramidal coder with quantization noise feedback. Each step of the pyramid constructs a high frequency difference signal consisting of the difference between the reconstructed quantized lowpass image from the pyramid one step above. The coder transmits the difference signals and one final lowpass representation of the image, which has been filtered and decimated many times. Most often, the number of pixels transmitted is more than in the original image, but because energy compaction occurs in the frequency decomposition, data compression is still achievable.

This thesis studies only the integration of subband filters with MC. The integration of pyramidal coders with MC is the topic of another research project.

Chapter 3

Performance and Statistical Measures

Codec performance can be measured objectively and/or subjectively. Objective measurements of video codecs are the easiest to make, but objective performance measures do not imply the same subjective performance rating. In subjective measurements, the human visual system (HVS) is used to perceive impairments. As a result, objective and subjective measures of the same sequences may have different or even reversed performance ratings. Most basic statistical measures calculate only first and second order statistics and are generally used to help estimate where the energy in the compressed images is located, to design predictors, and to estimate source entropy.

This chapter defines a number of objective performance and statistical measures.

3.1 Video Performance Measures

The most common objective performance measure used in image and video compression research is the peak-signal-to-noise-ratio (PSNR). This parameter is defined as

$$\text{PSNR} = 10 \log_{10} \left(\frac{256^2}{MSE} \right) , \quad (3.1)$$

where the numerator represents the square of the peak input pixel amplitude and the denominator is the mean squared error between original and reconstructed images. This measure is used to indicate the overall quality of the codec system's reconstructed

images. A rule of thumb states that the HVS can perceive a changes in image quality of 1 dB. Even though the PSNR measures the squared error, it does not tell where the error occurs in the frame and does not directly measure HVS perceivable impairments; it is, therefore, only a reasonable estimate of image quality.

The relationship between PSNR, quality, and the number of bits per pixel, R , is the critical characteristic of a codec. The PSNR is calculated as above, but R is measured either by counting encoded bits or by using a statistical estimate. As defined in Chapter 2, the zeroth order entropy is defined as

$$H = - \sum_{j=1}^L p_j \log_2 p_j \quad , \quad (3.2)$$

where p_j represents the probability of symbol j occurring in the symbol sequence. The zeroth order function considers each symbol independently, whereas higher order entropy measures consider groups of symbols and result in a lower entropy value if the symbols are correlated.

3.2 Video Statistical Measures

Simple first and second order image statistics can be estimated using the sample data if we make the assumption that the measured video data is stationary or wide-sense-stationary.

Consider a video sequence, Ω , consisting of a number of image frames, Ω_n , with spatial width and height of $X \times Y$ pixels respectively, and pixel amplitudes indexed as $\Omega_n(x, y)$. Then, the frame mean μ_{Ω_n} is defined as

$$\mu_{\Omega_n} = \frac{1}{XY} \sum_{x=1}^X \sum_{y=1}^Y \Omega_n(x, y) \quad , \quad (3.3)$$

and the frame variance $\sigma_{\Omega_n}^2$ is defined as

$$\sigma_{\Omega_n}^2 = \frac{1}{XY} \sum_{x=1}^X \sum_{y=1}^Y (\Omega_n(x, y) - \mu_{\Omega_n})^2 \quad . \quad (3.4)$$

The better the encoder, the more redundancy removed from the source. One statistical measure, called correlation, is used to indicate how much redundancy has been

removed. The source symbol to symbol correlation generally decreases for better encoders. Assuming a stationary case, the spatial one-step correlation can be estimated from the covariance function as follows. The spatial covariance function $r(m, n)$ (Jain 1989) for a frame is defined as

$$r(x, y) = \frac{1}{XY} \sum_{x'=1}^{X-x} \sum_{y'=1}^{Y-y} (\Omega_n(x', y') - \mu_{\Omega_n}) (\Omega_n(x + x', y + y') - \mu_{\Omega_n}) \quad . \quad (3.5)$$

Using this function, the one-step correlations in the x and y directions are defined as

$$\rho_x = \frac{r(1, 0)}{\sigma_{\Omega_n}^2} \quad (3.6)$$

and

$$\rho_y = \frac{r(0, 1)}{\sigma_{\Omega_n}^2} \quad . \quad (3.7)$$

The spatial covariance function can be used to construct the covariance matrix, as defined in Eq. 2.11, when designing an affine predictor.

The temporal covariance function, relative to frame n , is defined as

$$r(z) = \frac{1}{XY} \sum_{x'=1}^X \sum_{y'=1}^Y (\Omega_n(x', y') - \mu_{\Omega_{n,n-z}}) (\Omega_{n-z}(x', y') - \mu_{\Omega_{n,n-z}}) \quad , \quad (3.8)$$

where $\mu_{\Omega_{n,n-z}}$ represents a two frame mean. The temporal correlation is calculated by normalizing (3.8), notably

$$\rho_z = \frac{r(z)}{\sigma_{\Omega_{n,n-z}}^2} \quad , \quad (3.9)$$

where $\sigma_{\Omega_{n,n-z}}$ is estimated using pixel amplitudes from both temporal frames. The temporal covariance is used to measure the correlation between two frames.

Chapter 4

MC - Subband Filtering Video Codec Design

This chapter shows how to design and build video codecs that integrate MC with subband filtering tools. This is followed by a discussion of expected codec performance ranking. Included in the design process are quantizer design and bit allocation algorithms. A discussion of these algorithms is also presented. In this work, the goal is to study the performance of video codecs that integrate MC and subband filtering, but practical codecs have other components, such as quantizers and entropy coders, that also affect the system's performance. Therefore, a generic codec is proposed for those coding operations common to all systems studied. A description of a basic video codec is given below.

Practical video codecs cascade coding tools in a certain order, with each tool performing some form of data compression or transformation. Often, this ordering has a significant impact on the codec's performance. Consider Figure 4.1, which shows a basic codec. The quantization and entropy coding steps have been separated from the main encoder functional block for reasons described below. This codec is symmetric

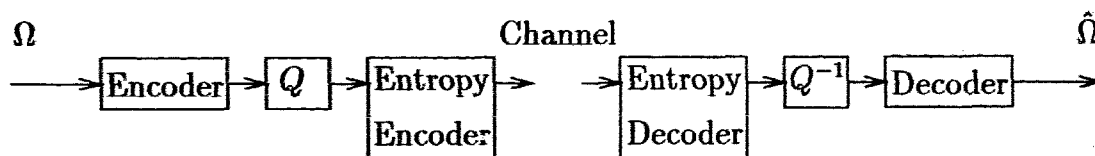


Figure 4.1: A Symmetric Video Codec Configuration

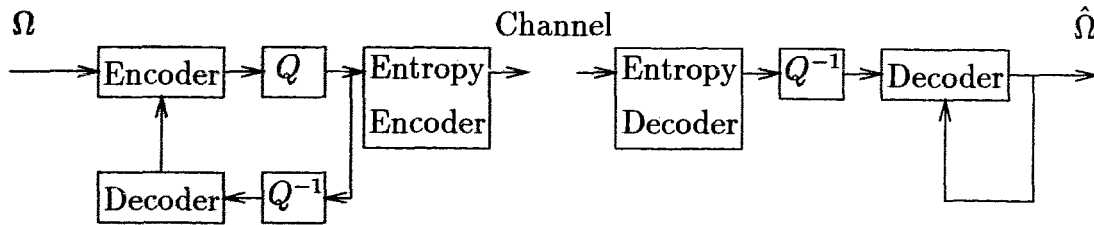


Figure 4.2: A Video Codec with Feedback

in configuration, whereas Figure 4.2 shows an alternative non-symmetric configuration that requires decoder feedback. Both configurations are shown, because MC and DPCM coding tools require feedback, while others, such as subband filtering and transform coders, do not. The symmetric codecs studied here have lower computational loads than those of non-symmetric design. The non-symmetric system encoders have much higher computational complexities than their respective decoders. In these figures, the encoder/decoder blocks represent the integrated coding tools that have been studied in this work.

In order to keep as many as possible codec variables constant, the quantizer and entropy coders are set apart from the encoder/decoder functional blocks and made the same for all systems developed. The idea is to make a generic quantizer-entropy codec block that is representative of an actual system. Uniform quantizers were used instead of pdf optimized quantizers, because their performance when cascaded with entropy encoders is similar to pdf optimized quantizers and their implementation is simplified. They are also typically used in video coding standards. In the simulations, the entropy coder bit rate was estimated using the zeroth-order-entropy measure.

Given a particular system configuration, further design questions remain, such as how to assign bits among n sources and how to design the uniform quantizers for a particular total data rate. Further in this chapter, descriptions on how to integrate MC and subband filtering video coding tools, how to design uniform quantizers, and how to perform bit allocations are given. Expected codec performance rankings are also discussed.

4.1 Video Sequence Notation

The notation for the video frame sequences is important for understanding the following discussions. This notation is summarized below.

Let MC, spatial subband filtering, and temporal subband filtering operations be denoted by M, S, and T respectively. Let the encoder input and reconstructed decoder output video sequences be denoted by Ω and $\hat{\Omega}$ respectively, and let frame i in these sequences be denoted by Ω_i and $\hat{\Omega}_i$. For MC systems, let the video sequence of DFD frames be denoted by Ω_{DFD} , let the set of motion vectors for frame i be denoted by MV_i , and the complete sequence of motion vectors be denoted by MV . Similarly, for T and S systems, the subband sequences are denoted by Ω_T and Ω_S . For subband filtered sequences, low-pass l 's and/or high-pass h 's will be added to the notation when describing specific subbands. For example, the video sequence notated by Ω_{T-S}^{l-h} represents the temporally low-pass, horizontally low-pass, and vertically high-pass filtered subband in a TS codec. lastly, let $\tilde{\Omega}$ represent a quantized video sequence.

4.2 MC and Subband Filtering Codecs

There are many ways to integrate MC with subband filtering. Some are relatively easy to implement while others are not. Much of the complexity occurs when placing coding tools into the MC feedback loop, especially with temporal subband filtering. This section describes how to apply MC, spatial subband filtering, and temporal subband filtering individually, both pair-wise, and in triples. If one counts all the combinations implied above, fifteen different systems are obtained: three individual, six pair-wise, and six triplet configurations. First we will study the M, S, and T codecs; then the TS, ST, SM, MS, TM, and MT codecs; and finally, the TSM, STM, TMS, SMT, MST, and MTS codecs. In addition, four modified systems called SM1, TM1, TSM1, and STM1 are studied.

The output video sequence(s) from each system's encoder is quantized and entropy coded. Uniform quantization is used directly on all sequences except the special case described next. When subband filtering without using MC, the lowest frequency subband generally has frame statistics similar to those of the original frame. Because of this, DPCM is generally used to remove remaining spatial correlations when

encoding these bands (Woods and Naveen 1992). Therefore, to simulate a generic quantizer codec operation, a DPCM with in-loop uniform quantizer is used to quantize low-passed subbands in codecs that use only subband filtering. The entropy of the quantized video sequences is estimated using the zeroth-order-entropy instead of actual codec implementations. For those systems using M, the motion vector entropies were estimated similarly.

4.2.1 Codecs Using One Video Coding Tool

Individually, the three coding methods, M, S, and T, have been conceptually introduced in Chapter 2. Described below is the implementation of each of these methods.

The implementation of an M codec for our video codec system is straightforward. Refer to Figure 4.3, which shows the video sequences and motion vector sequences created in the encoding and decoding MC process. The i th frame is represented by the symbol Ω_i in the middle of a vertical line. In addition, the i th set of motion vectors is represented by the symbol MV_i underneath the angled arrow. The encoder transforms the input video sequence, Ω , into the video sequence Ω_{DFD} and motion vectors sequence MV . Each Ω_{DFD_i} and MV_i is constructed by applying the MC algorithm to the present frame Ω_i and using the reconstructed frame $\hat{\Omega}_{i-1}$ as the search frame. Following this, MV_i and the quantized Ω_{DFD_i} , $\tilde{\Omega}_{DFD_i}$, are entropy coded and transmitted. The decoder reconstructs the sequence $\hat{\Omega}$ from $\tilde{\Omega}_{DFD}$ and MV . For each time step at the receiver, the arriving information is entropy decoded and split into frame $\tilde{\Omega}_{DFD_i}$ and the corresponding MV_i . Using the previously reconstructed frame $\hat{\Omega}_{i-1}$, frame $\tilde{\Omega}_{DFD_i}$ and MV_i , frame $\hat{\Omega}_i$ can be reconstructed. This reconstruction process also occurs in the encoder's feedback decoder.

The implementation of a four-band spatial subband filtering codec is symmetric, as was shown pictorially in Figures 2.10 and 2.11. For each Ω_i , the S encoder constructs four images representing information from different frequency bands in the two-dimensional space. Because of decimation, each subband image, Ω_{S_i} , is one-quarter the size of the original input frame Ω_i . The subband frames at time step i are denoted as $\Omega_{S_i}^{ll}$, $\Omega_{S_i}^{lh}$, $\Omega_{S_i}^{hl}$, and $\Omega_{S_i}^{hh}$. Quantization of the four frequency bands produces the video sequences $\tilde{\Omega}_{S-DPCM}^{ll}$, $\tilde{\Omega}_S^{lh}$, $\tilde{\Omega}_S^{hl}$, and $\tilde{\Omega}_S^{hh}$, which are entropy coded and transmitted. The receiver constructs the output video sequence $\hat{\Omega}$ by decoding

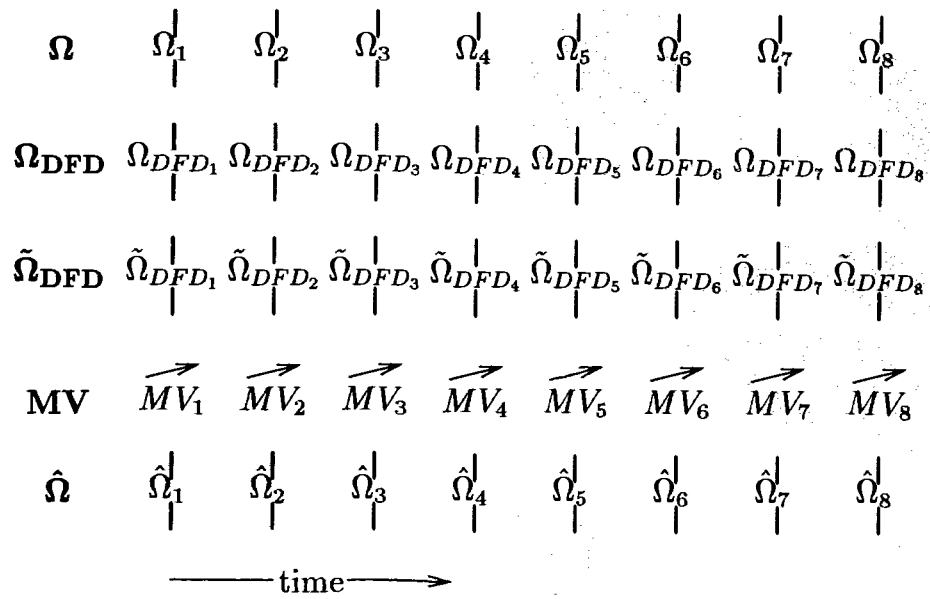


Figure 4.3: The Coding of a Video Sequence using Motion Compensation

the subbands and synthesis subband filtering.

The implementation of a temporal subband filtering codec is also symmetric, but the decoded frames are time-shifted by a delay equal to the filter set reconstruction delay. This adds complexity to all systems that use the T coding tool. To show this, consider Figure 4.4, which depicts the frame sequences created when a temporal subband filtering system is used to code a video sequence of eight frames. In the figure, Ω , Ω_T^l , Ω_T^h , $\tilde{\Omega}_{T-DPCM}^l$, $\tilde{\Omega}_T^h$, and $\hat{\Omega}$ represent respectively the original, temporally low-pass, temporally high-pass, temporally low-pass DPCM encoded, temporally high-pass quantized, and reconstructed video sequences. In addition, the filter set has a filter reconstruction delay of one time-step. First, the input video sequence Ω is both low-pass and high-pass temporally filtered to construct the Ω_T^l and Ω_T^h video sequences. The Ω_T^l and Ω_T^h sequences are then down-sampled by removing the odd frames. The remaining even-number frames are quantized, producing the sequences $\tilde{\Omega}_{T-DPCM}^l$ and $\tilde{\Omega}_T^h$, and entropy coded. Following this, the decoder builds the $\tilde{\Omega}_T^l$ and $\tilde{\Omega}_T^h$ sequences. Then the synthesis filters are used to construct the sequence $\hat{\Omega}$, but with a time-step delay equal to the filter set reconstruction delay. In this example, a delay of one occurs. A disadvantage of this technique is that the decoder must store a number of previous time-step low- and high-pass filtered images in order to be able

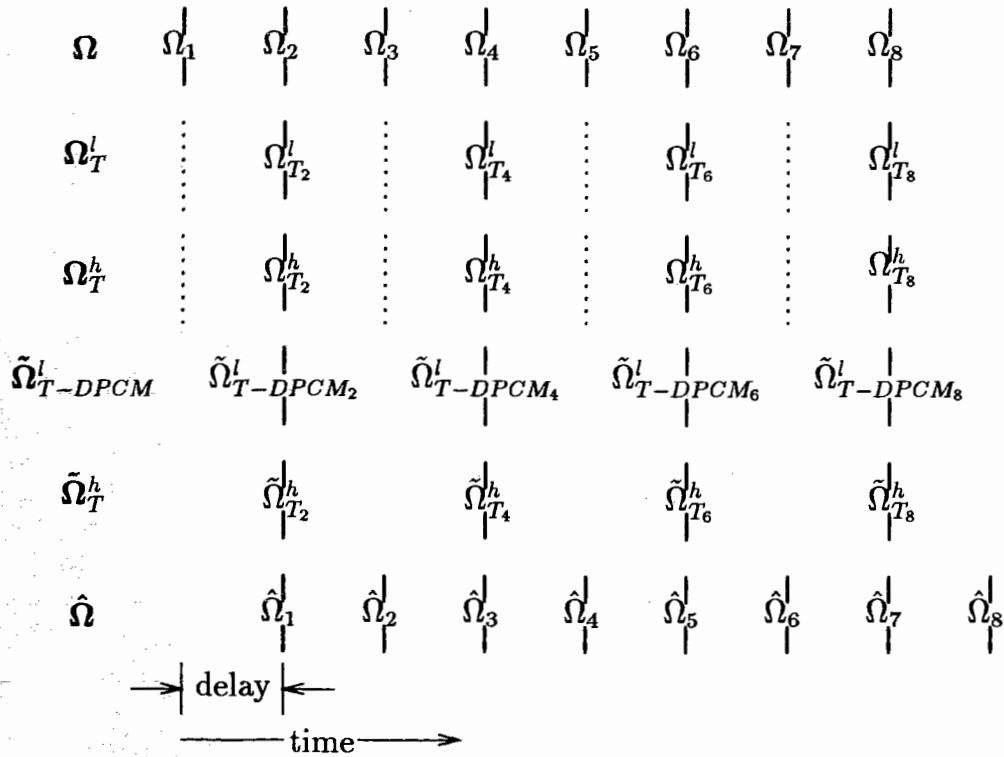


Figure 4.4: A Temporally Filtered Video Sequence

to do the required synthesis temporal filtering. The number of frames required for both high and low pass subband sequences is $n = F \text{ modulo } 2 + F/2$, where F is the number of taps in the longest filter in the filter set. The filter delay also adds an implementation complexity at codec startup. One can either pad the video with blank frames or use circular filtering. For practical systems, the padding method would be chosen, whereas circular filtering may be used for codec simulations. The padding method was used in this work. In addition to codec startup, the temporal filter codec performance may degrade during scene changes because of changing video statistics. Depending on how much the video statistics change between scenes, the codec may perform better if the codec is reset at these times.

4.2.2 Codecs Using Two Video Coding Tools

Video codecs using pairs of the M, S, and T coding tools are more complex to implement. The six possible configurations are TS, ST, SM, MS, TM, and MT. The design of each of these systems, as well as for two modified systems called SM1 and TM1,

are given below.

The configuration of a TS codec system is similar to a T system, but with the addition of further S coding of the Ω_T^l and Ω_T^h subband sequences. As a result, eight subband sequences are created: Ω_{T-S}^{l-ll} , Ω_{T-S}^{l-lh} , Ω_{T-S}^{l-hl} , Ω_{T-S}^{l-hh} , Ω_{T-S}^{h-ll} , Ω_{T-S}^{h-lh} , Ω_{T-S}^{h-hl} , and Ω_{T-S}^{h-hh} . Continuing with the same notation as in Figure 4.4, only the even frames in these sequences are further encoded, again because of the temporal down-sampling in the T encoder. The eight subbands are quantized and entropy encoded. The receiver reconstructs the eight subbands and then synthesis TS filters them to form the output video sequence $\hat{\Omega}$. The output sequence is also time-shifted by a delay equal to the filter set reconstruction delay.

The configuration of an ST codec system is straightforward, and can be easily implemented. In this system, the input video Ω is S encoded, producing the four subband sequences Ω_S^{ll} , Ω_S^{lh} , Ω_S^{hl} , and Ω_S^{hh} . Next, each of the four subband sequences are simply encoded with a T encoder as if they were ordinary input sequences. This system, and all systems that use both S and T coding tools, will produce eight subbands. The symbols used for the eight subbands are: Ω_{S-T}^{ll-l} , Ω_{S-T}^{ll-h} , Ω_{S-T}^{lh-l} , Ω_{S-T}^{lh-h} , Ω_{S-T}^{hl-l} , Ω_{S-T}^{hl-h} , Ω_{S-T}^{hh-l} , and Ω_{S-T}^{hh-h} . These subbands are quantized and entropy coded. The receiver reconstructs the subbands and synthesis filters the subbands resulting in the output video sequence $\hat{\Omega}$. Because separable filters are used, the performance of the TS and ST systems is expected to be very similar, if not the same.

The SM codec configuration is simple and easy to implement. Here, the input sequence Ω is S encoded into four S sequences. Then each of the S sequences, Ω_S^{ll} , Ω_S^{lh} , Ω_S^{hl} , and Ω_S^{hh} , are separately M encoded into the DFD frame video sequences Ω_{S-DFD}^{ll} , Ω_{S-DFD}^{lh} , Ω_{S-DFD}^{hl} , and Ω_{S-DFD}^{hh} and motion vector sequences MV_S^{ll} , MV_S^{lh} , MV_S^{hl} , and MV_S^{hh} . The four DFD frame sequences are quantized before they and their respective motion vector sequences are entropy coded. The receiver performs entropy decoding, M decoding on each of the four data streams, and then S synthesis filtering to construct the output video sequence $\hat{\Omega}$.

The system SM1 (Paek, Kim, and Lee 1992) is a modified version of an SM codec. The modification is that M is performed only on the subband sequence Ω_S^{ll} and not on the other subband sequences. This system would transmit the encoded video sequences Ω_{S-DFD}^{ll} , Ω_S^{lh} , Ω_S^{hl} , and Ω_S^{hh} and the motion vector sequence MV_S^{ll} . This system is of interest, because it requires only one quarter the number of MC

computations compared with the SM system. Its performance may be hypothesized to be similar to that of the SM system.

The configuration of an MS system consists of the basic structure of an M system, but with the addition of S coding the DFD frame video sequence Ω_{DFD} . Because the S codec is placed after the M system, only one motion vector sequence MV is created. On the other hand, four subbands are input to the quantizer. The subband sequences are denoted as: Ω_{DFD-S}^{ll} , Ω_{DFD-S}^{lh} , Ω_{DFD-S}^{hl} , and Ω_{DFD-S}^{hh} . Each quantized subband sequence and motion vector sequences are entropy coded and transmitted. The decoder performs inverse entropy coding, reconstruction of the subbands, S synthesis filtering, and M decoding to construct the output video sequence $\hat{\Omega}$.

The configuration of a TM system is similar to that of an SM system. Instead of applying M to the four S subbands in an SM system, M is applied to the T subband video sequences Ω_T^l and Ω_T^h in a TM system. For the two parallel M coders, the respective quantized video sequences $\tilde{\Omega}_{T-DFD}^l$ and $\tilde{\Omega}_{T-DFD}^h$, and the motion vector sequences MV_T^l , MV_T^h , are entropy coded and transmitted. The output video sequence in the receiver is again the sequence $\hat{\Omega}$. The search frame used by the MC algorithm is two time-steps back, since the T sequences are decimated by two.

The system TM1 is a modified version of a TM codec. The modification is that M is performed only on the subband sequence Ω_T^l and not on the Ω_T^h subband sequences. This system would transmit the encoded video sequences $\tilde{\Omega}_{T-DFD}^l$ and Ω_T^h , and the motion vector sequence MV_T^l . Like the SM system, this system is also of interest, because it requires only one-half the number of MC computations compared to the TM system. Its performance is hypothesized to be similar to that of the TM system, especially if the high-pass bands contain little information.

In this work, the most complex coding configuration to implement is that of when an M coder precedes a T coder. Two reasons make this system complex. First, temporal decimation implies that the MC algorithm must use search frames that are modulo-2 frames back instead of the previous frame. Second, the temporal reconstruction delay forces an even larger temporal delay between the encoded frame and the search frame. To explain why this is so, consider the two pictorial frame sequences shown in Figures 4.3 and 4.4; if the figures are overlaid so that the M sequence Ω_{DFD} represents the T system's input sequence Ω , then the complexity can be seen. To make this simpler to understand, Figure 4.5 shows this overlapping when coding frames Ω_i

and Ω_{i-1} . At time step i , the decoder output frame sequence is delayed by τ time steps, where τ is equal to the filter set reconstruction delay. Therefore, the search frame used to encode frames i and $i - 1$ is the reconstructed output frame $\hat{\Omega}_{i-1-\tau}$. In the figure, the two dotted vertical lines point to this frame. This output frame is the most recent frame that can be reconstructed at time-step i and $i - 1$. The figure also shows the temporal video sequences Ω_T^l and Ω_T^h that are quantized to the $\tilde{\Omega}_T^l$ and $\tilde{\Omega}_T^h$ sequences that, in turn, are entropy coded. The sequence $\hat{\Omega}_{DFD}$ represents the reconstructed DFD sequence in the synthesis filter bank of the decoder. In this system, both the two quantized temporal bands and the motion vector information are entropy coded and transmitted. Because of the filter delay, buffering of the motion vector sequence must occur in either the encoder or decoder until the delayed DFD frames are reconstructed and decoded. This system has a major disadvantage in that the M algorithm must search temporally delayed frames instead of the previous frame. One would expect that, as the filter delay increases, the codec performance will degrade.

These six orderings comprise all pair-wise combination of the three coding tools M, S, and T. The pair-wise configuration descriptions will be used when describing the triple-wise configurations below.

4.2.3 Codecs Using Three Video Coding Tools

The eight video codec configurations using all three coding tools M, S, and T are labelled TSM, STM, TSM1, STM1, TMS, SMT, MTS, and MST. For all codecs, quantization and entropy coding are performed on all transmitted image sequences, while only entropy coding is applied to the motion vector sequences. Because all these systems use M coding, DPCM quantization is not used in any of these systems.

The first two systems, TSM and STM, are the simplest configurations of the six triple-coding-tool systems. Each system is constructed by applying M to the eight subband sequences of a TS or ST coder. The eight separate DFD frame sequences and the motion vector sequences are transmitted.

The next two systems, TSM1 and STM1, are modified versions of the TSM and STM systems respectively. In these systems, M is applied only to the lowest frequency subband sequences Ω_{T-S}^{l-l} and Ω_{S-T}^{h-l} for the systems TSM1 and STM1 respectively.

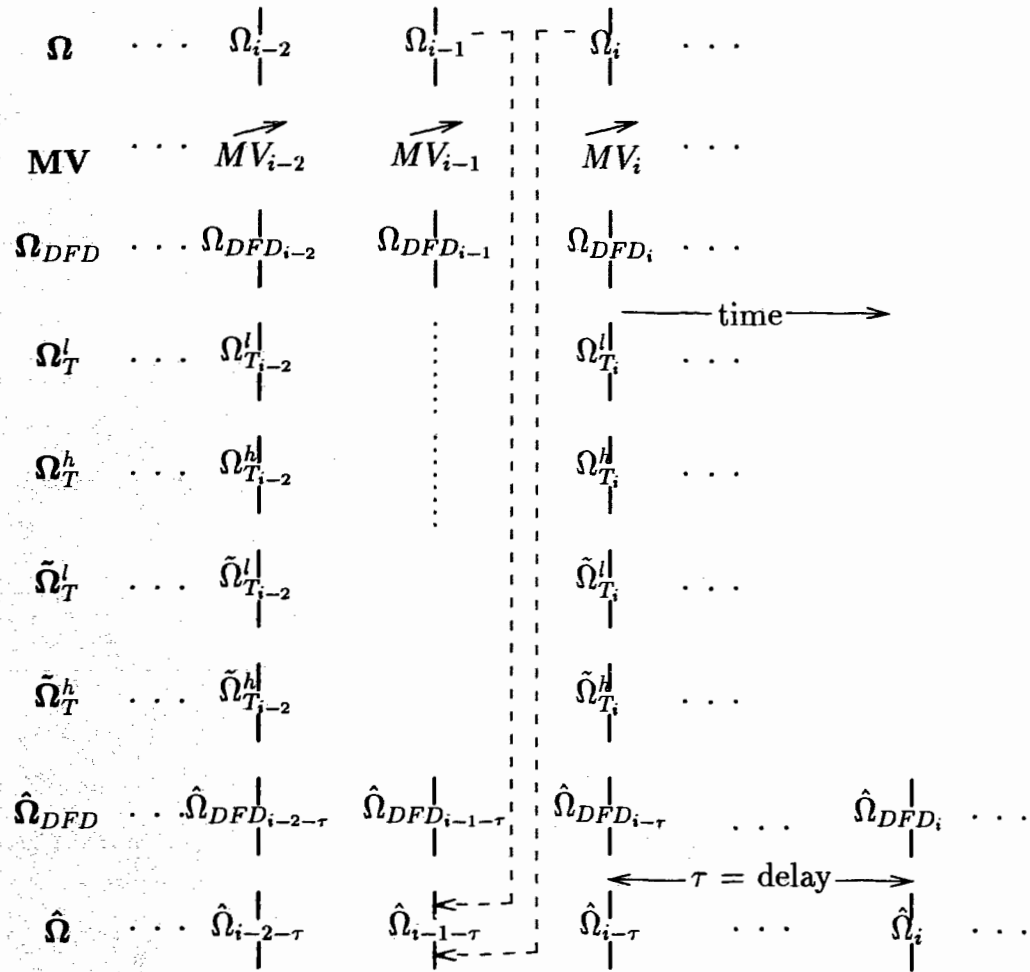


Figure 4.5: An MC - Temporally Filtered Video Sequence

The remaining seven subband sequences are not M coded. The reason for studying these systems is the same as that given for the SM1 and TM1 systems. It is hypothesized that the TSM1 and STM1 systems will perform similarly to the TSM and STM systems, especially if the high-pass bands contain little information.

The configuration of a TMS system can best be seen by considering the configuration of a T system cascaded with an MS system. Two motion vector sequences \mathbf{MV}_T^l and \mathbf{MV}_T^h are constructed and transmitted, whereas, eight subband-DFD sequences $\Omega_{T-DFD-S}^{l-ll}$, $\Omega_{T-DFD-S}^{l-lh}$, $\Omega_{T-DFD-S}^{l-hl}$, $\Omega_{T-DFD-S}^{l-hh}$, $\Omega_{T-DFD-S}^{h-ll}$, $\Omega_{T-DFD-S}^{h-lh}$, $\Omega_{T-DFD-S}^{h-hl}$, and $\Omega_{T-DFD-S}^{h-hh}$, are constructed and transmitted.

Similarly, the configuration of an SMT codec can be seen as the cascading of

an S codec and an MT codec. The four output sequences of the S coder are input to four separate MT codecs. In this system, four motion vector sequences and eight subband-DFD video sequences are constructed and transmitted: MV_{S-DFD}^{ll} , MV_{S-DFD}^{lh} , MV_{S-DFD}^{hl} , and MV_{S-DFD}^{hh} , and the subband-DFD video sequences are denoted as: $\Omega_{S-DFD-T}^{ll-l}$, $\Omega_{S-DFD-T}^{ll-h}$, $\Omega_{S-DFD-T}^{lh-l}$, $\Omega_{S-DFD-T}^{lh-h}$, $\Omega_{S-DFD-T}^{hl-l}$, $\Omega_{S-DFD-T}^{hl-h}$, $\Omega_{S-DFD-T}^{hh-l}$, and $\Omega_{S-DFD-T}^{hh-h}$.

The last two systems, MTS and MST, are similar in configuration. These systems place the three-dimensional subband codecs, TS and ST inside the M feedback loop. These systems are subject to a reconstruction frame delay, due to the temporal filtering operator. Only one motion vector sequence is created in these systems but, as usual, eight DFD-subbands are created. The notation used for the subbands is the same as that for the TS and ST systems except that a “DFD” is added to the subscript.

4.3 Expected Codec Performance Rankings

The expected performance rankings of the nineteen MC-subband filtering codecs described in the previous section are hypothesized here. In addition to the nineteen systems, three more systems, PCM, DPCM, and MDCT, are added to the rankings as representative standard coding methods. In the discussion below, a “-” is used to represent the same or comparable ranking.

Starting with the standard systems, the ranking in descending order is expected to be M – MDCT, DPCM, and PCM. The M codec is expected to perform similarly to an MDCT system, but the MDCT has improved strengths during scene changes and at low rates, because of the DCT’s abilities at coding still images. A DPCM codec is expected to perform worse than MC based systems and better than a PCM codec, because the MC predictive coding performs better and PCM does not remove redundancy.

Next, S, T, TS, and ST systems are considered. The ranking in this group in descending order is expected to be TS – ST, S, and T. The performance of the TS and ST codecs is expected to be similar, especially if separable filters are used. Because TS and ST operate in both temporal and spatial dimensions, they are expected to perform better than either S and/or T codecs. Two-dimensional spatial subband filtering is

expected to perform better than one-dimensional temporal subband filtering, because of the added dimension and the increase in the number of subbands in the codecs designed in this work. It is hypothesized that the subband filtering codecs will rank below the M and MDCT codecs, but above the DPCM and PCM codecs, because of MC abilities to remove temporal redundancy and subband filtering energy compaction abilities respectively.

Now consider the SM, SM1, TM, TM1, MS, and MT codecs. The expected ranking in descending order for these systems is SM, SM1 – MS – TM, and TM1 – MT. SM coding is expected to have the best ranking here, since S coding performs better than T and because T and M coding tools remove only temporal redundancies in TM based codecs. Both temporal and spatial redundancies are removed in SM based codecs. The performance of all modified systems, such as SM1 and TM1, is expected to be below the respective systems, in this case, SM and TM. The performance of MS is expected to be below SM because S filtering of the DFD frames places more energy into the high frequency subbands as compared with the SM subbands. This occurs because the DFD frames contain proportionally more high frequency information than the original frames, i.e., the DFD frames contain information in the regions of motion where poor prediction estimates are made and the frames tend to have a zero-mean. As a result, subband filtering after MC does not achieve the same amount of energy compaction as compared to when it is used before MC. The same is expected when comparing the MT and TM systems. Comparing these system performances to the previous two groups, the SM codec is expected to perform similarly to the M – MDCT systems. The worst performing codecs in this group, TM1 – MT, are expected to perform comparably to the TS – ST codecs.

A last grouping of codec configurations include the TSM, STM, TMS, SMT, MTS, MST, TSM1, and STM1 systems. The expected ranking in descending order for these systems is TSM – STM, SMT – TSM1 – STM1, and TMS – MTS – MST. The TSM and STM codecs are expected to perform similarly, since TS and ST are expected to have the same performance. The codecs that place MC last in the configurations are expected to perform better than those that place it earlier, because subband filtering of the DFD frames is hypothesized to not achieve as much energy compaction as subband filtering of the original video frames (see discussion in previous paragraph). Again, the modified systems, TSM1 and STM1, are expected to perform worse than

TSM and STM respectively. It is not known how well they will perform compared to the other systems, but it is hypothesized they will perform comparably with the SMT system. The TSM – STM codecs are expected to rank similarly with or just below the SM and M – MDCT codecs. These systems are expected to perform even better than the SM and M – MDCT codecs if the video has little information in the high frequency subbands, because most of the video information will be compacted in the low frequency subbands. The TMS – MTS – MST codecs are expected to rank similarly to the MS, MT, and TS – ST systems.

4.4 Uniform Quantizer Design Methods

The design of a uniform quantizer for a given source is generally a function of the input statistics. As described in Chapter 2, a uniform quantizer can be either midtread or midrise, can have a dead-zone or not, and has a parameter called the step-size (Δ). For the sources in this work, the pdf's tend to be Laplacian in shape, highly peaked at zero (Karlsson and Vetterli 1988a; Woods and O'Neil 1986). For example, Figure 4.6 shows the 12th DFD frame from the video test sequence "Ping Pong" and the DFD frames histogram. The pixels in the frame are scaled by 4 and offset by 127 so the DFD image detail can be seen in the figure. Midtread quantizers with a reconstruction output value at zero tend to perform better for these sources than midrise types, because the high density of low amplitude values around zero dominate the quantizer distortion.

If simple midtread or midrise uniform quantizers are to be designed, the design

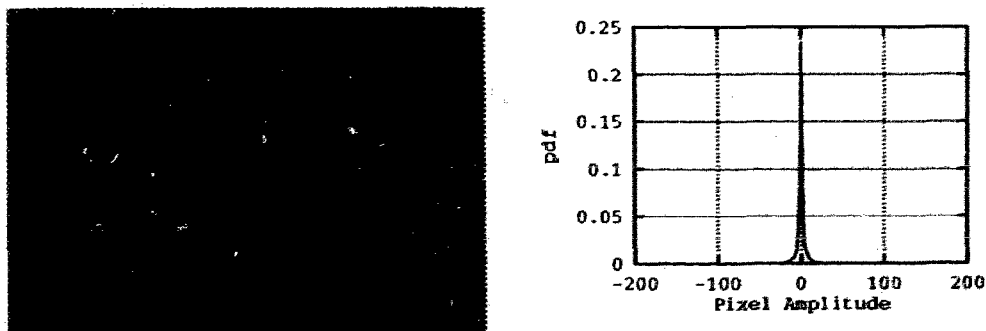


Figure 4.6: An MC DFD Frame and PDF From the "Ping Pong" Video Sequence

parameters are the step-size (Δ) and the bit rate (R), which translates directly into the number of reconstruction values $L = 2^R$. For a given bit rate (R), the Δ that minimizes the distortion between the quantizer input and reconstructed values is desired. Figure 4.7a shows this relationship. For small Δ , the distortion (D) is high and is dominated by the overload distortion effects, while for large Δ , D is dominated by granular noise. Between these two extremes the distortion has a minimum that represents the optimal Δ for the given R . Procedures exist for estimating Δ from the source variance σ^2 (Jayant and Noll 1984), but they do not guarantee an optimal value. If a training set with statistics representative of the source is obtained, minimization algorithms, such as the Golden Section search (Press et al. 1988), can be used. Besides the D versus Δ relationship, the quantizer function that relates D to R is useful when performing bit allocations among many sources. Bit allocation is described in the next section. Given a set of optimal Δ 's for given R 's, the D versus R relationship for a uniform quantizer is shown in Figure 4.7b. In a well-behaved system, the shape is convex. The higher the R , the lower the D .

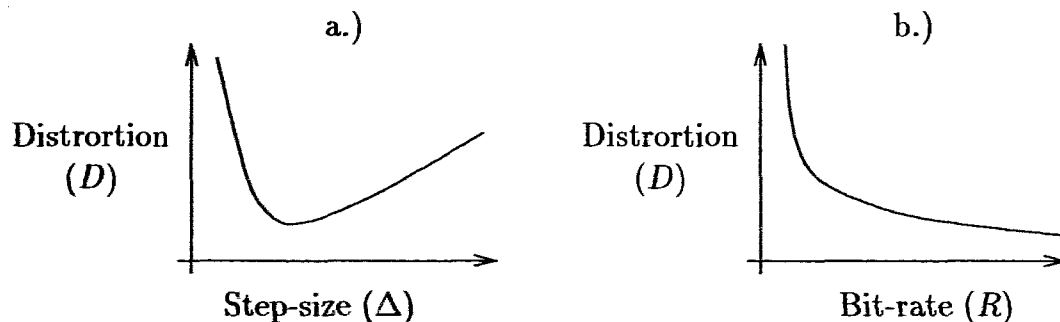


Figure 4.7: Uniform Quantizer a.) Distortion versus Step-size and b.) Distortion versus Bit Rate Relationships

4.5 Bit Allocation Methods

In addition to codec configuration and quantizer design concerns, the allocation of bits among several quantizers is an important factor in the systems performance. Given a fixed bit rate, how do we assign bits to the quantizers to achieve the best system performance? In simple MC systems that do not segment the DFD images, only one image sequence is quantized and transmitted; however, in subband/DCT systems,

more than one image/coefficient sequence is transmitted and bit assignment design problems exist. For the discussion in this section, let each different image sequences represent a data source. The bit allocation problem therefore becomes one of assigning bits among n sources.

Numerous bit allocation algorithms exist. Three common methods include the greedy algorithm (Gersho and Gray 1992), an analytical algorithm (Gersho and Gray 1992; Woods and Naveen 1992), and the Breiman, Friedman, Olshen, and Stone (BFOS) algorithm (Riskin 1991). The greedy algorithm incrementally assigns bits to the sources based on those that contribute the most distortion. The analytical algorithm simply calculates the bit allocation as a function of the source variances. The calculated allocations must be adjusted to non-negative integer rates, since the calculation produces fractional rates and sometimes even negative rates. This algorithm is best suited to sources that are created using the same coding methods. The BFOS algorithm, on the other hand, finds the optimal bit allocation using tree searches, and can easily be used for sources with varying statistics. This is the technique used in this work.

The BFOS algorithm works on the following principle. A distortion rate table is constructed by independently calculating quantizer noise power distortions at a number of rates for each source and then combining this information into a table. The algorithm uses this table to find a bit allocation for a user specified target rate. Consider Figure 4.8 which shows the relationship of average distortion to average rate where the dots represent all possible bit allocations using the independent source distortion rate table data. The lowest possible distortion for a given rate is described by the solid curve on the graph, called the convex hull; the points joined by the dotted lines represent optimal bit allocations for particular rates along the convex hull. The convex hull is not always attainable for all optimal bit allocations. In the figure, only four bit allocations are on the convex hull. The BFOS algorithm starts allocating bits at a high rate and then traces out the convex hull by deallocating bits until a bit allocation rate equal to or less than the target rate is reached. Only those allocations on the convex hull are selected by the BFOS algorithm; therefore, to select those optimal allocations not on the convex hull, the greedy algorithm is used to add bits to the BFOS bit-allocation until the target rate is reached. The specifics of the algorithm are given in Riskin (1991).

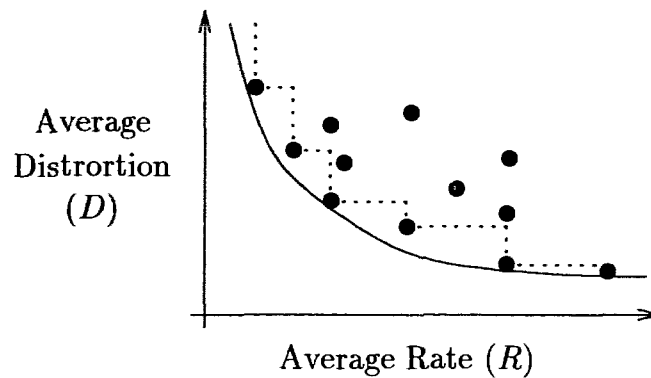


Figure 4.8: BFOS Distortion versus Rate Relationship

The distortion rate table can be constructed when designing the quantizers. This is straightforward when the source distortions have a one-to-one scaling or have the same weighting on the final reconstructed video. When this is not so, such as in subband filtering, the distortion must be scaled appropriately. The next section describes subband weighting factors.

4.5.1 A Filter Bank Noise Power Weighting Estimate

The mean-squared-error (MSE) between the original and the corresponding quantized frame represents the noise power. It is often useful to know how the quantizer noise power in each subband scales to the output, especially when performing quantizer bit allocations among the bands. The non-unity noise scaling results from the synthesis filter frequency responses. A derivation on how to weight subband noise power through one-, two-, and three-dimensional synthesis filter banks follows. Three assumptions are used here: the subbands are independent of each other, one-dimensional separable filters are used, and the noise power is white.

Consider the single synthesis subband filtering step shown in Figure 4.9, where the input signal $x(n)$ is up-sampled and filtered to get the output signal $y_2(n)$. Using

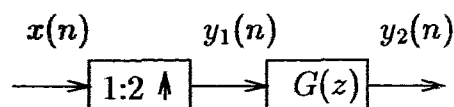


Figure 4.9: A Subband Synthesis Filter

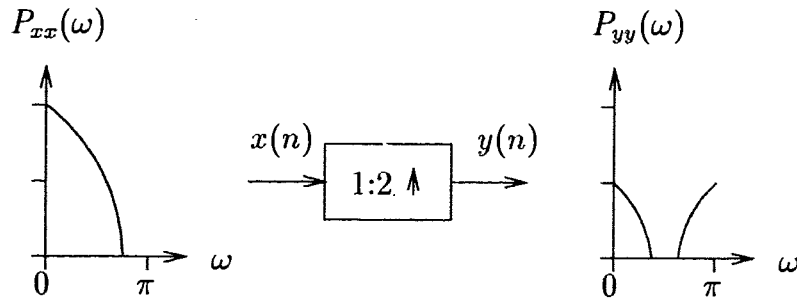


Figure 4.10: Spectral Imaging of Interpolation Operator

basic stochastic process theory for linear-shift-invariant (LSI) systems, Woods and Naveen (1992) have shown that the average power spectral density of the output of an up-sampler is

$$P_{yy}(\omega) = \begin{cases} \frac{1}{2}P_{xx}(2\omega), & 0 \leq \omega \leq \frac{\pi}{2} \\ \frac{1}{2}P_{xx}(2\pi - 2\omega), & \frac{\pi}{2} \leq \omega \leq \pi \end{cases} . \quad (4.1)$$

This relationship is shown pictorially in Figure 4.10. Next, the relationship between the input, x , and output, y , for a LSI system, such as the filter $G(z)$, is given by

$$P_{yy}(\omega) = |H(\omega)|^2 P_{xx}(\omega) . \quad (4.2)$$

Using these two relationships, an expression that estimates the contribution of the noise from the input to the output can be found. Ignoring the source information for the moment, let $x(n)$ represent the noise signal and assume the noise is white with variance σ_x^2 . This assumption is good for a high bit rate uniform quantizer. Then, the input power spectral density is

$$P_{xx}(\omega) = \sigma_x^2 , \quad (4.3)$$

and, using (4.1) and (4.2), the power spectral densities of $P_{y_1y_1}$ and $P_{y_2y_2}$ are

$$P_{y_1y_1}(\omega) = \frac{\sigma_x^2}{2} \quad (4.4)$$

and

$$P_{y_2y_2}(\omega) = \frac{\sigma_x^2}{2} |G(\omega)| . \quad (4.5)$$

The variance $\sigma_{y_2}^2$ of the output is of interest. Using Parseval's Theorem,

$$\sigma_{y_2}^2 = \overline{R_{y_2y_2}}(0) = \frac{1}{2\pi} \int_{-\pi}^{\pi} P_{y_2y_2}(\omega) d\omega = \frac{\sigma_x^2}{4\pi} \int_{-\pi}^{\pi} |G(\omega)|^2 d\omega = \frac{\sigma_x^2}{2} \sum_{-\infty}^{\infty} |g(n)|^2 . \quad (4.6)$$

From this relationship, the scaling of the noise through the filter bank is

$$w = \frac{1}{2} \sum_{-\infty}^{\infty} |g(n)|^2 \quad . \quad (4.7)$$

When subband filtering two-dimensional and three-dimensional data with separable one-dimensional subband filters, the noise power scaling is simply the product of the individual weighting factors, thus, for separable two-dimensional filters,

$$w_k = w_k^h w_k^v \quad , \quad (4.8)$$

where w_k^h , and w_k^v represent the vertical and horizontal weighting factors respectively. Similarly, with separable three-dimensional filters,

$$w_k = w_k^h w_k^v w_k^t \quad , \quad (4.9)$$

where w_k^h , w_k^v , and w_k^t represent the vertical, horizontal, and temporal weighting factors respectively. These weighting factors are used to scale the subband distortion values in the BFOS algorithm's distortion rate table. The algorithm then assigns bits as a function of the reconstructed noise power, as desired.

Chapter 5

Video Codec Simulations and Results

In the previous chapters, video coding tools, performance measures, and video codec configurations were described. Using this information, simulations of twenty-two video codecs were run and their performances recorded.

This chapter describes the simulation test video sequences, the subband filtering filter sets, and the video codec simulations and results.

5.1 Test Video Sequences

Three standard eight bit precision monochrome video test sequences are used in this work. The three sequences represent different scenes and differing levels of motion. The sequences, “Miss America”, “Ping Pong”, and “Salesman”, are labeled “*missa*”, “*pongi*”, and “*sales*” respectively in this thesis. The *missa* sequence has low-motion and shows a person’s head and shoulders shot with a low detailed background. The *pongi* sequence shows a high-motion ping-pong game, whose scene pans right and then left. Finally, the *sales* sequence has medium-motion and shows a salesman, seated at a desk, talking and moving his arms. In *sales*, the background has high detail. The first frames in each of these sequences are shown in Figure 5.1.

All three video sequences have frame widths of 360 pixels, but were changed to a width of 356 pixels so that the MC macro blocks of size 8×8 were evenly divisible into the spatially down-sampled subbands. The height of the frames in *missa* and

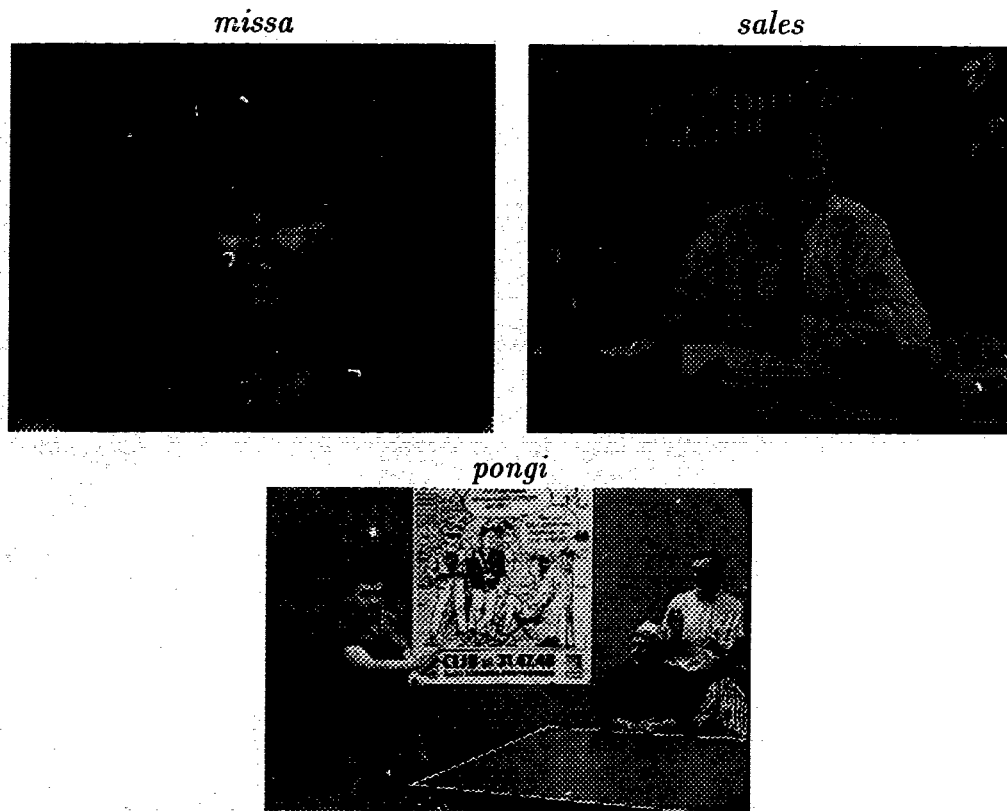


Figure 5.1: First Frames of Simulation Test Video Sequences *missa*, *sales*, and *pongi*

sales is 288 pixels, and 240 pixels in *pongi*. In all simulations, the first 30 sequence frames were used.

5.2 Subband Filtering Filter Sets

Seven different filter sets were used in this work, since their choice also affects system performance. The sharper the cutoff region in the frequency response, the lower the aliasing energy in the subbands, which one would expect to result in a lower bit rate. Also, quadrature mirror filter sets produce uncorrelated subbands, because the filter sets are orthogonal to each other, but this feature does not say anything about the correlation inside the subbands. If the decorrelation is significant, the performances might show this. On the other hand, short kernel filters require fewer computations and may have a performance similar to longer filters of other design. In order to

explore these issues, sample quadrature mirror filters (QMF), conjugate quadrature filters (CQF), and short kernel perfect reconstruction filters (PRF) were studied. Three filters were of the QMF type: a 2 tap filter defined by Smith and Barnwell (1986); and the 16b and 32c filters designed by Johnston (1980). These filters will be denoted as the 2-QMF, 16b-QMF, and 32c-QMF filters respectively. The two CQF filters are the 8 and 16 tap filters designed by Smith and Barnwell (1986). These filters will be denoted as the 8-CQF and 16-CQF filters respectively. The last two filters were the 3-5 and 4 tap perfect reconstruction filters (PRF) designed by LeGall and Tabatabai (1988) and are denoted as the 3-5-PRF and 4-PRF filters respectively.

Figure 5.2 shows the frequency responses for all seven filter sets. For each filter set, the analysis and synthesis low and high pass filters responses are shown. The impulse responses for each filter set are given in Appendix B. As the figure indicates, the longer the impulse length, the flatter the in-band response and the sharper the cutoff region. The longer filters have a more ideal frequency response; however, they require extra computations to implement. For all filter sets, the analysis filters have unity gain and the synthesis filters have a gain of two. In the subband filtering process, a gain of two is required to restore the original signal power after up-sampling.

The filter set weighting factors for one- and two-dimensional subband filtering are given in Tables 5.1-5.2. Other than the weighting factors for the 3-5-PRF and 4-PRF filter sets, the factors have unity values to four decimal places. These factors are used to scale the subband image quantizer noise powers so that the BFOS bit allocation algorithm assigns bits based on the reconstructed noise power.

Table 5.1: One-Dimensional Weighting Factors for the Seven Filter Sets

Filter	Weighting Factors	
	w_l	w_h
2-QMF	1.0000000000	1.0000000000
3-5-PRF	0.7500000000	1.4375000000
4-PRF	0.6250000000	2.5000000000
8-CQF	1.0000001192	1.0000001192
16-CQF	1.0000052452	1.0000051260
16b-QMF	1.0000545979	1.0000545979
32c-QMF	1.0000630597	1.0000630597

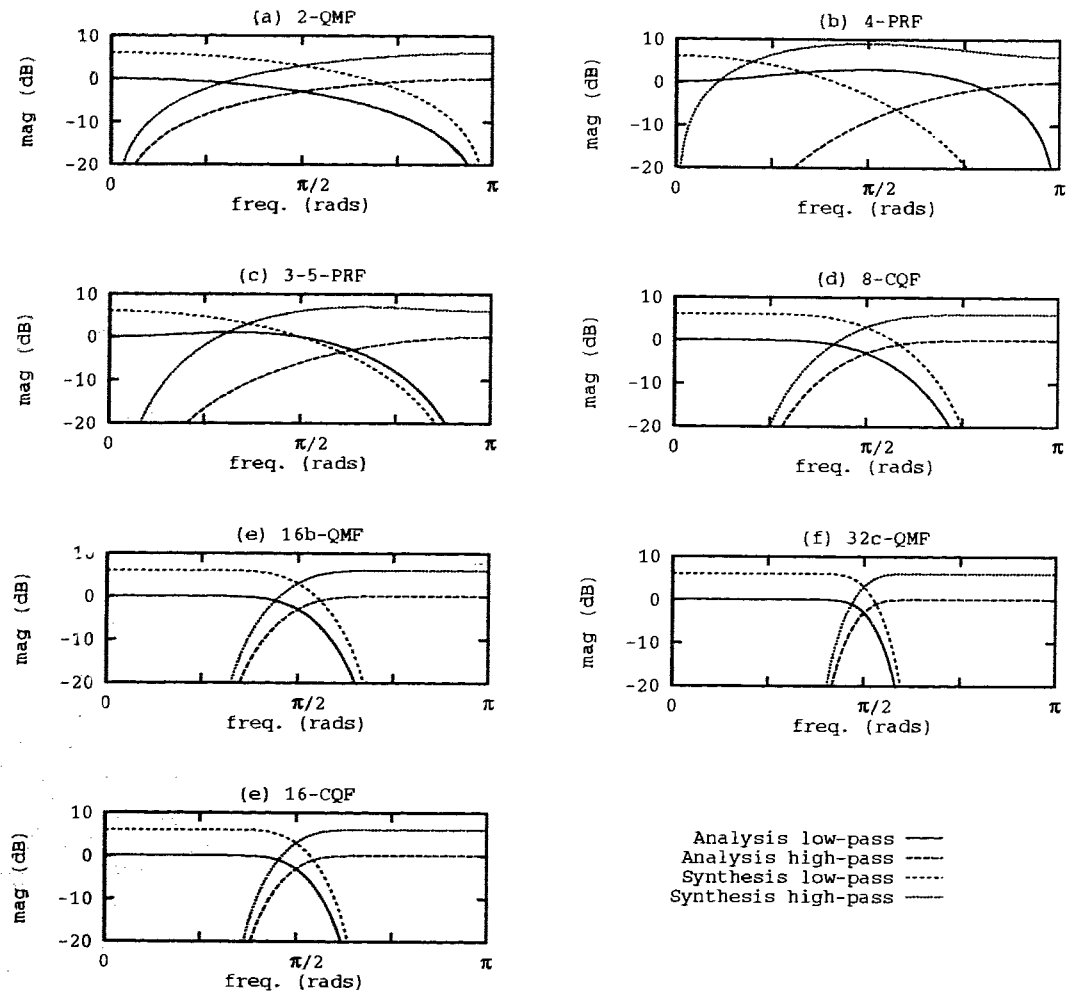


Figure 5.2: The Seven Filter Sets Frequency Responses

5.3 Simulations and Results

Video codec simulations were performed using the test sequences and filter sets described above. In fact, a total of twenty-two different video codecs were constructed and tested. The systems include PCM, DPCM, and MDCT configurations in order to have a baseline against which to compare the MC/subband filtering codecs. The MDCT configuration is an M codec that uses the DCT to further code the DFD blocks. This codec was developed as a representative of standard systems such as MPEG and $p \times 64$. The remaining system configurations consist of the fifteen systems and the four modified systems detailed in the last chapter. The aim of this section is to present codec results, and to show the strengths and weaknesses of each

codec.

The video codec performance results using the three sequences *pongi*, *missa* and *sales* will be presented for all the systems. The major performance measurement used is the relationship between the PSNR and the entropy (bit rate). It is desired to have the highest PSNR for the lowest bit rate possible. The simulations varied the codec bit rate over the range of 0-3 bits/pixel. The second performance measure recorded is the set of correlation coefficients, ρ_x , ρ_y , and ρ_z . These values were measured before quantization and are used as an indication of the codec's ability to remove redundancy. Finally, the third performance measure studied is the encoded pixel variances. If the variance is reduced, the bit rate is expected to decrease.

In all systems, the Golden Section search was used for designing the uniform midtread quantizers, and the BFOS algorithm for bit assignments. The MC block sizes were chosen to the standard size of 8×8 and the conditional full search was used. The conditional full search transmits the present macro block if its energy is lower than any of its corresponding DFD blocks calculated using the full search algorithm. The search window size parameter p was set to 8 when encoding full size frames and set to 4 when encoding spatially decimated subband frames. The search variable was halved to keep the search region in the S codec subband frames, which were decimated by two in each spatial direction, the same as in the full sized frames. The block size for the spatially and temporally decimated frames remained at 8×8 . The MSE distortion measure was used to find the block matches.

The results are presented in four major groupings. The standard PCM, DPCM, M, and MDCT systems are presented as Group 1. Single and pair-wise systems S, T, TS, MS and MT performances are presented individually and are then compared as a group against the M and MDCT performances; these codecs define Group 2.

Table 5.2: Two-Dimensional Weighting Factors for the Seven Filter Sets

Filter	Weighting Factors			
	w_{ll}	w_{lh}	w_{hl}	w_{hh}
2-QMF	1.0000000000	1.0000000000	1.0000000000	1.0000000000
3-5-PRF	0.5625000000	1.0781250000	1.0781250000	2.0664062500
4-PRF	0.3906250000	1.5625000000	1.5625000000	6.2500000000
8-CQF	1.0000002384	1.0000002384	1.0000002384	1.0000002384
16-CQF	1.0000104904	1.0000103712	1.0000103712	1.0000102520
16b-QMF	1.0001091957	1.0001091957	1.0001091957	1.0001091957
32c-QMF	1.0001261234	1.0001261234	1.0001261234	1.0001261234

In Group 3, the pair-wise systems SM, SM1, TM, and TM1 results are given and comparisons are made between them and the standard systems. In Group 4, triple systems STM, STM1, TMS, SMT, and MTS results are given. All the systems are ranked in order of performance, and the best performing systems are then discussed and compared. Finally, subjective comparisons are made of three systems, MDCT, SM, and TSM. It was found that the performance of systems using a TS or ST grouping was very similar since separable filters were used. As a result, only the TS grouping results are being presented. This applies to the ST, STM, STM1, and MST systems.

Before objective performance results are presented, a word about the subjective meaning of the PSNR quality measure is given here. The higher the PSNR value, the better the reconstructed video quality, but the PSNR scale shifts for different video sequences. At 1 bit/pixel, a PSNR of 33, 42, and 41 dB for the *pongi*, *missa*, and *sales* sequences respectively is considered a good system performance. Subjective quality evaluations are made of images shown in Figures 5.24 and 5.25.

5.3.1 Video Codec Results for Group 1: PCM, DPCM, M, and MDCT

To start, consider the standard coding systems PCM, DPCM, M, and MDCT and their performance. Figure 5.3 shows the PSNR versus entropy relationships for these systems. As expected, the PCM system is inferior to the other systems, which all remove redundancies between pixels. For all three test sequences, the DPCM codec performs much better than the PCM but poorer than the M and MDCT systems. The MDCT codec performs best in *missa*, and *sales*, and for low rates in *pongi*. The MDCT system performance curves extend to lower rates than for the other systems. This occurs here because there are many more sources to produce fractional rates, i.e., 64 DCT coefficient sources. The same could be achieved for the other systems if the images were to be segmented into a number of sources. A source can be divided into N sources by using an assignment law to assign the samples to the sources. One would generally want all pixels in a given source to share a common property (eg: background or foreground). Source segmentations were not done here, because the systems perform well for the bit rates of interest. Table 5.3 shows the temporal and

Table 5.3: Group 1 Correlation and Weighted Variance Video Statistics

sequence	codec	σ^2	ρ_x	ρ_y	ρ_z
pongi	PCM	2713.431	0.8471	0.8181	0.7289
	DPCM	465.435	0.0191	0.0751	-0.1408
	M	114.441	0.1958	0.4391	-0.0433
missa	PCM	872.747	0.9868	0.9751	0.9921
	DPCM	15.725	-0.3407	-0.0650	-0.0875
	M	6.547	0.0762	0.3790	-0.0410
sales	PCM	1148.700	0.9268	0.9305	0.9842
	DPCM	55.704	-0.1579	0.0348	0.3540
	M	15.572	0.1965	-0.0104	-0.3554

spatial one step correlation coefficients and variances for the PCM, DPCM, and M systems. The values indicate that, as the system performance increases, the pixel correlations decrease and the variances drop. These results are not given for the MDCT case, because the DCT coefficients are usually coded block-by-block instead of coefficient-by-coefficient at the frame level. The MDCT correlation coefficients are important; however, the correlation coefficients used here show redundancy removal in the subbands, and the MDCT codec is used only as a benchmark system with which to compare overall performances.

There is a cost to increased performance: computational load. As an indication of this load, the number of multiplications and additions per frame are estimated for each system. A PCM codec adds no computational load, whereas the DPCM predictor adds multiplications and additions. Given frame widths and heights of $W \times H$ pixels and using the three step predictor defined in Chapter 2, the computational load for a DPCM encoder is

$$\begin{aligned} \Gamma_m &= 3WH \quad \text{per frame} \\ \Gamma_a &= 3WH \quad \text{per frame} \end{aligned} \quad (5.1)$$

where Γ_m and Γ_a represent the number of multiplications and additions. Using block sizes of $N \times N$, a search window size parameter p , the full search method, and the

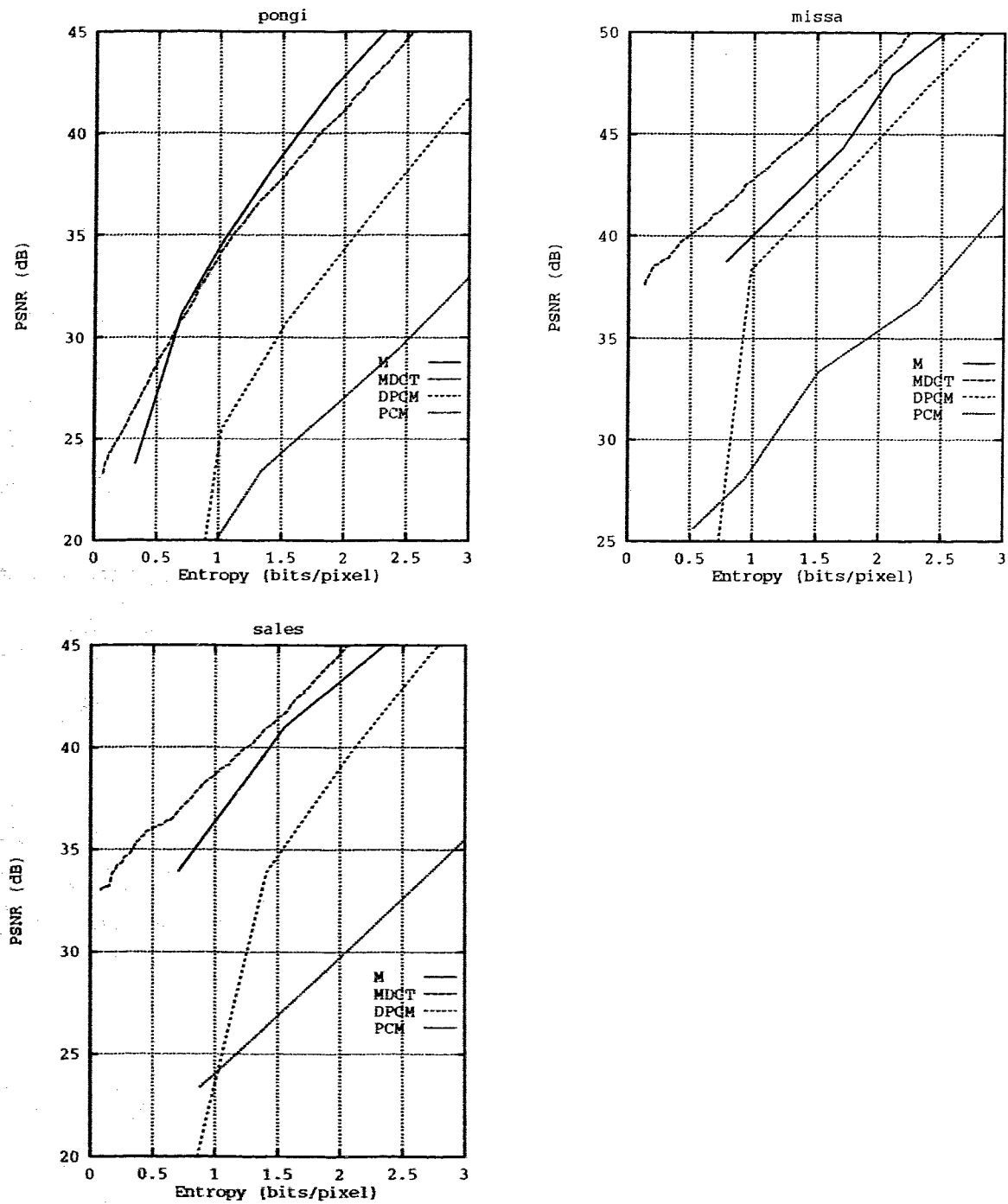


Figure 5.3: PSNR versus Entropy for Group 1 Systems: PCM, DPCM, M, and M-DCT

MSE distortion measure, the computational load per frame for an M encoder is

$$\begin{aligned}
 \Gamma_m &= \left(\frac{\# \text{ mults}}{\text{block compare}} \right) \left(\frac{\# \text{ block compares}}{\text{encoded block}} \right) \left(\frac{\# \text{ encoded blocks}}{\text{frame}} \right) \\
 &= (N^2 + 1)(2p + 1)^2 \left(\frac{WH}{N^2} \right) \text{ per frame} \\
 \Gamma_a &= \left(\frac{\# \text{ adds}}{\text{block compare}} \right) \left(\frac{\# \text{ block compares}}{\text{encoded block}} \right) \left(\frac{\# \text{ encoded blocks}}{\text{frame}} \right) \\
 &= (2N^2 - 1)(2p + 1)^2 \left(\frac{WH}{N^2} \right) \text{ per frame}
 \end{aligned} \tag{5.2}$$

Note: the Γ_m and Γ_a per block compare is dependent on the distortion measure, and the number of block compares per encoded block is dependent on the search method. If the ABS distortion measure is used, multiplication is not required and, if the three step search method is used, the number of block compares per encoded block drops to $9 \log_2 p$ (Gothe and Vaisey 1993). The MDCT codec implements the recursive DCT algorithm developed by Hou (1987). For this algorithm, 63 multiplications and 183 additions are required per encoded block when $N = 8$. Using these formulas, the computational load measured in multiplications/additions for the DPCM, M, and MDCT systems to encode a macro block of size 8×8 pixels is 192/192, 18785/36703, and 18848/36886 computations respectively. The DCT encoding adds a load of 63/183 multiplication/additions per block to the M encoder. From these numbers, it is apparent that full search MC has an extremely high computation load. In a real M system, not all blocks in the frame are transmitted if conditional replenishment methods are employed. In this case, DCT computations are not required for un-transmitted blocks.

5.3.2 Video Codec Results for Group 2: S, T, TS, MS, and MT

Next, results for S, T, TS, MS, and MT are given. Figures 5.4-5.10 record the PSNR versus entropy relationship. For each system, several filter sets were used to determine the effect the filter sets have on the codec performance. The trade-off between performance and computations is important, so it is hoped that codecs using short kernel filter sets perform similarly or better than those codecs using longer kernel filter sets.

Figure 5.4 shows the PSNR versus entropy relationship for an S system and for all three video test sequences. In the figure, results for all seven filter sets are plotted. In the *pongi* sequence at low rates, below 1 bit/pixel, the performance of all the filters, except the 4-PRF filter, is very similar. At high rates, the 3-5-PRF filter has superior performance over the other filter sets. In the *missa* sequence, the 16b-QMF and 32c-QMF filters have the best performance at both low and high rates, but the 3-5-PRF filter's performance is not much worse than that of these filters. In the *sales* sequence, at low rates, all the filters except the 2-QMF and 4-PRF have similar results and, at high rates, the 3-5-PRF has the best performance followed by the 16b-QMF and 32c-QMF filter sets. These results show that no one filter set has the best performance for spatial subband filtering of video frames, and that performance is dependent on frame statistics. The results show also that the performance of the 3-5-PRF filter set has the best performance on average for all three sequences. This result is of interest, because of the short kernel length of the filter. It takes only one-eighth the computations to use the 3-5-PRF filters as compared to the 32c-QMF filters.

The work by Woods and Naveen (1992), comparing the performance of subband filter sets, showed that for low bit rates, 0.8–1.8 bits/pixel, the 16b-QMF and 32c-QMF performed better than the 3-5-PRF. The results here agree with that finding for the *missa* sequence at all rates and for *sales* at rates below 1 bit/pixel, but the *pongi* sequence results do not agree. This discrepancy is explained by the fact that Woods and Naveen's work was performed on different test sequences and that 16 subbands were used versus the 4 subbands used here. Even so, the results here show that the filter set performance is highly dependent on the video source and that the 3-5-PRF is a good choice for spatial subband filtering. The validity of this choice has been confirmed by Karlsson and Vetterli (1988a), who used the 3-5-PRF filter set because these filters have linear phase, low computational complexity, and relatively good frequency selection and interpolation properties.

Figure 5.5 shows the PSNR versus entropy relationship for a T system with all three video test sequences. In the figure, results for six filter sets are plotted. The filter set performance ratings differ when compared with filter set performance rankings in the S system. In the *pongi* sequence, the 2-QMF filter set has the best performance, followed by the 3-5-PRF filter set. In the *missa* sequence, the 2-QMF filter set has the

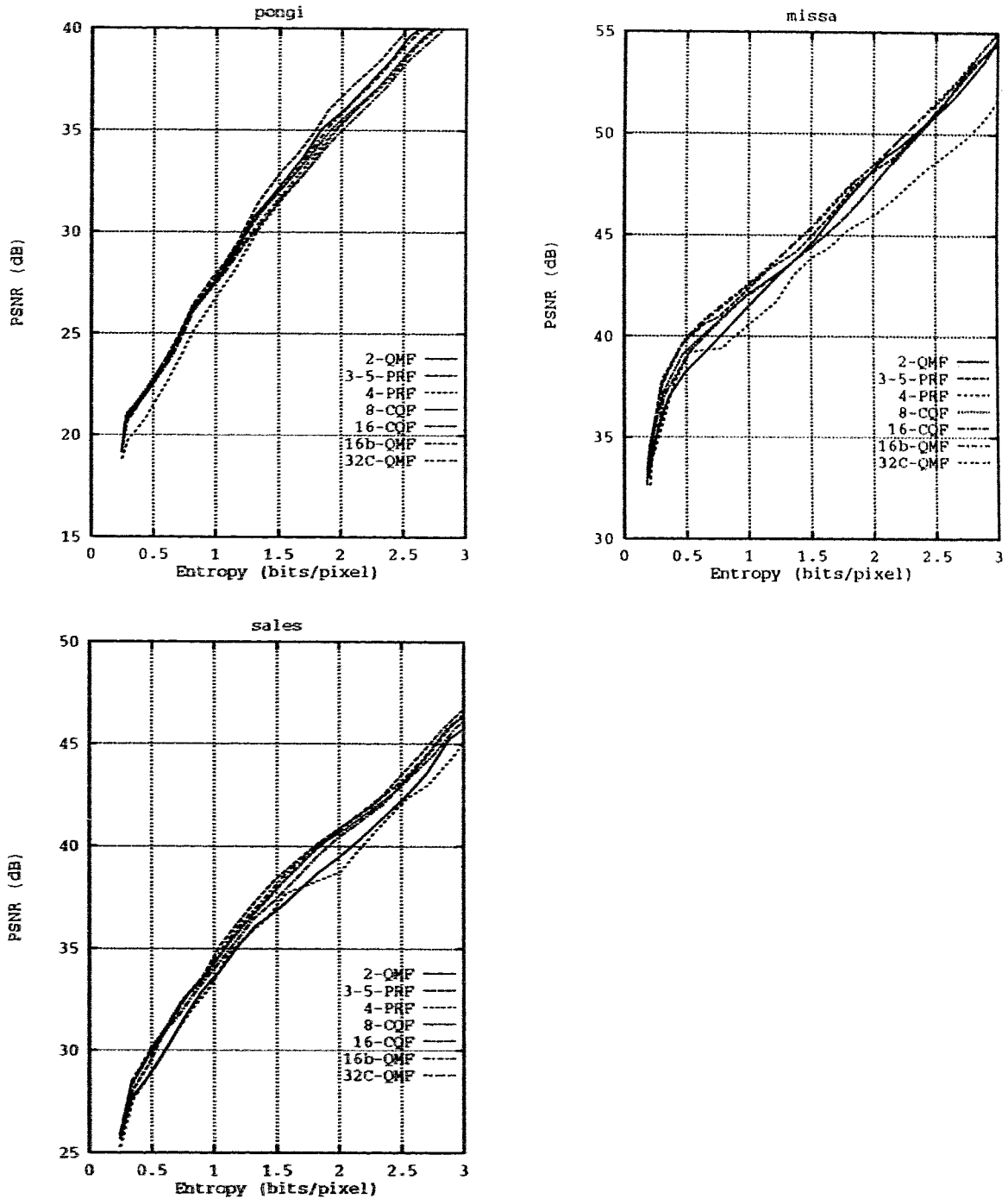


Figure 5.4: PSNR versus Entropy for the S System

best performance while the 16-CQF and 3-5-PRF filter sets have the next best. In the *sales* sequence, the 3-5-PRF filters have the best performance at low rates, and the 2-QMF filters have the best performance at high rates. These results show that the use of the 2-QMF filter set is best in terms of both performance and computational load. At low rates, several filter sets perform similarly but, at higher rates, the 2-QMF filter set performs as much as 3 to 4 dB higher than the other filter sets. Karlsson and Vetterli (1988a) also used the 2-QMF filter for temporal filtering; however, they do not justify its use in terms of performance but only in terms of complexity.

The next system to be discussed is TS. In this system, two different filter sets can be used: one for temporal and the other for spatial filtering. Figures 5.6-5.8 show the PSNR versus entropy relationships for the three temporal filter sets 2-QMF, 3-5-PRF, and 8-CQF respectively. The spatial filter sets used are shown in each graph's key. Figure 5.6 shows the results when the 2-QMF temporal filters were used. For *pongi*, the 3-5-PRF spatial filters performed best, followed by the 16b-QMF and 8-CQF filters. For *missa*, the 16b-QMF and 32c-QMF spatial filters performed best, but the 3-5-PRF filter also performed well at low rates. For *sales*, the 3-5-PRF spatial filter performed best at low rates while the 16b-QMF and 32c-QMF performed best at high rates. Next, Figure 5.7 shows the results when the 3-5-PRF temporal filters were used. For *pongi*, again the 3-5-PRF spatial filters performed best, followed by the 16b-QMF and 8-CQF filters. For *missa*, the 16b-QMF and 32c-QMF spatial filter sets performed best at low rates, the 16-CQF at high rates. For *sales* at low rates, all but the 2-QMF spatial filter performed the same, and the 16b-QMF and 32c-QMF performed best at high rates. Lastly, Figure 5.8 shows the results when an 8-CQF temporal filter set was used. The best spatial filter set was the 3-5-PRF for *pongi*, the 16b-QMF and 32c-QMF for *missa*, and the 3-5-PRF for *sales*. The best temporal/spatial filter set combination was the 2-QMF/3-5-PRF filter sets for *pongi*, and the 2-QMF/16b-QMF filter set for *missa* and *sales*. For each temporal filter set, the system performance for most of the spatial filter sets differs only, at the most by 1 dB in magnitude. More specifically, the 2-QMF/3-5-PRF performed less than 1 dB below all the best ranking filter sets. Therefore, favoring the benefits of lower computations, the temporal/spatial filter set choice of 2-QMF/3-5-PRF was considered the best for the TS system.

The MS and MT system results are discussed here and shown in Figure 5.9. In the

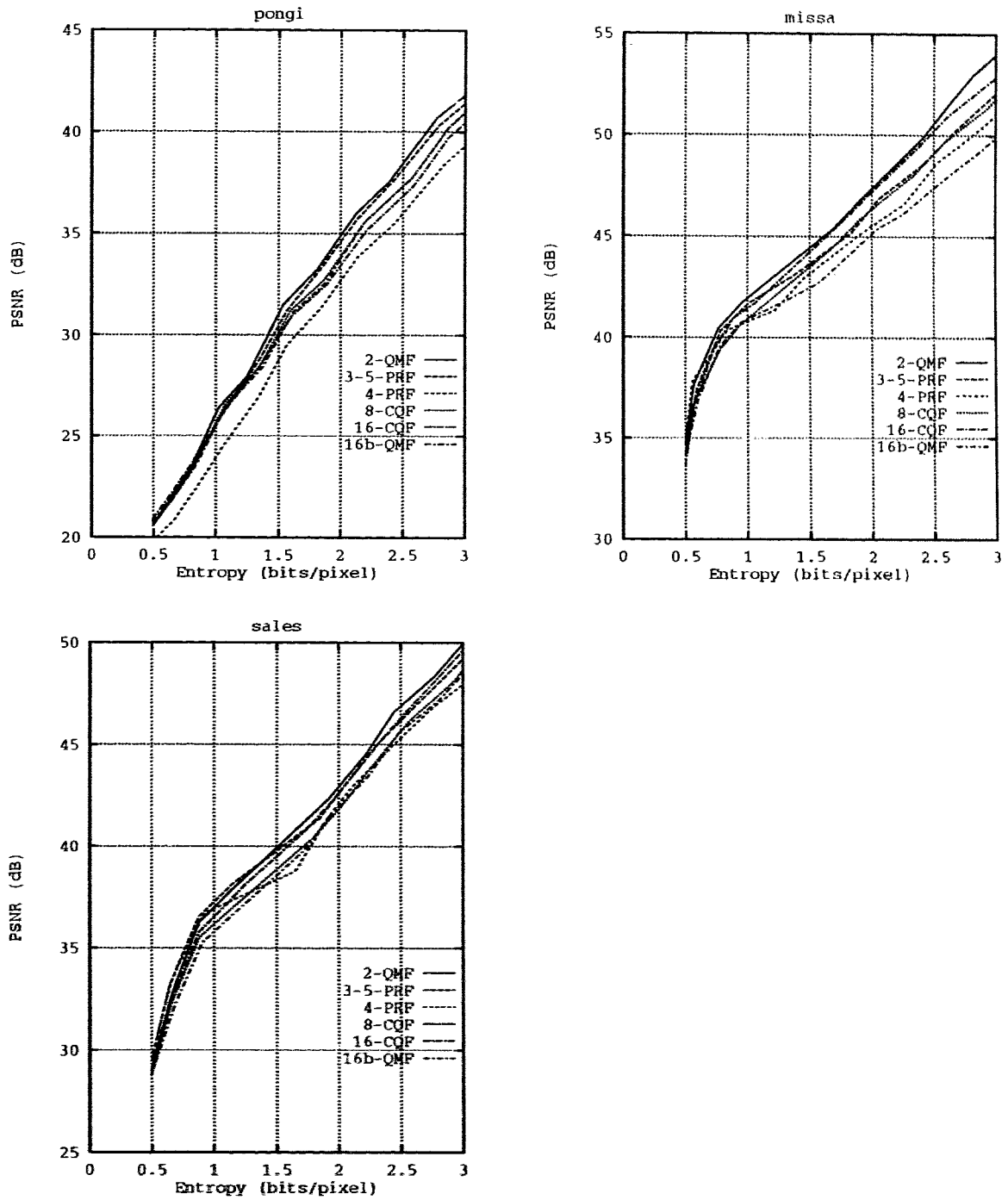


Figure 5.5: PSNR versus Entropy for the T System

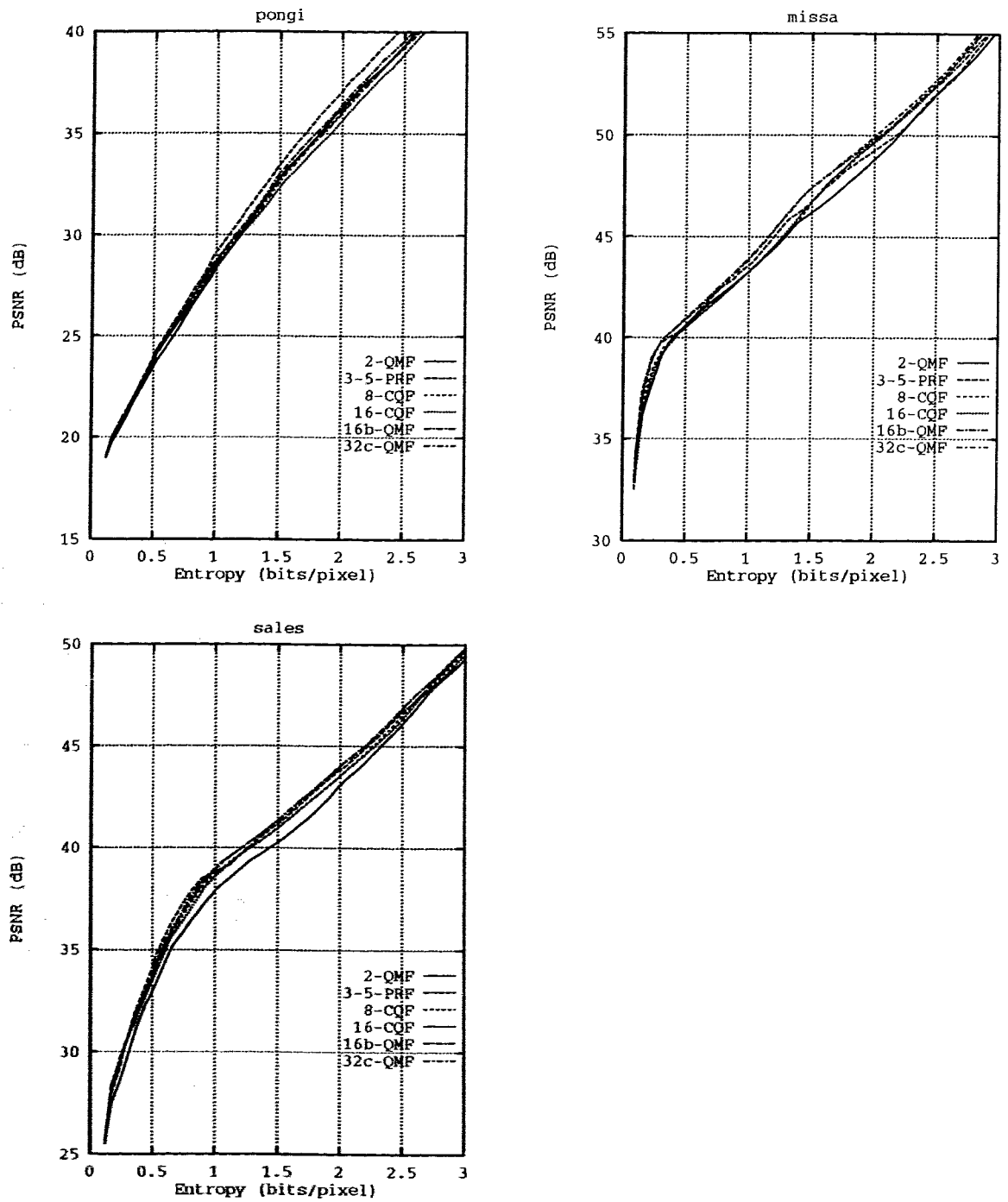


Figure 5.6: PSNR versus Entropy for the TS System with the 2-QMF Temporal Filter Set

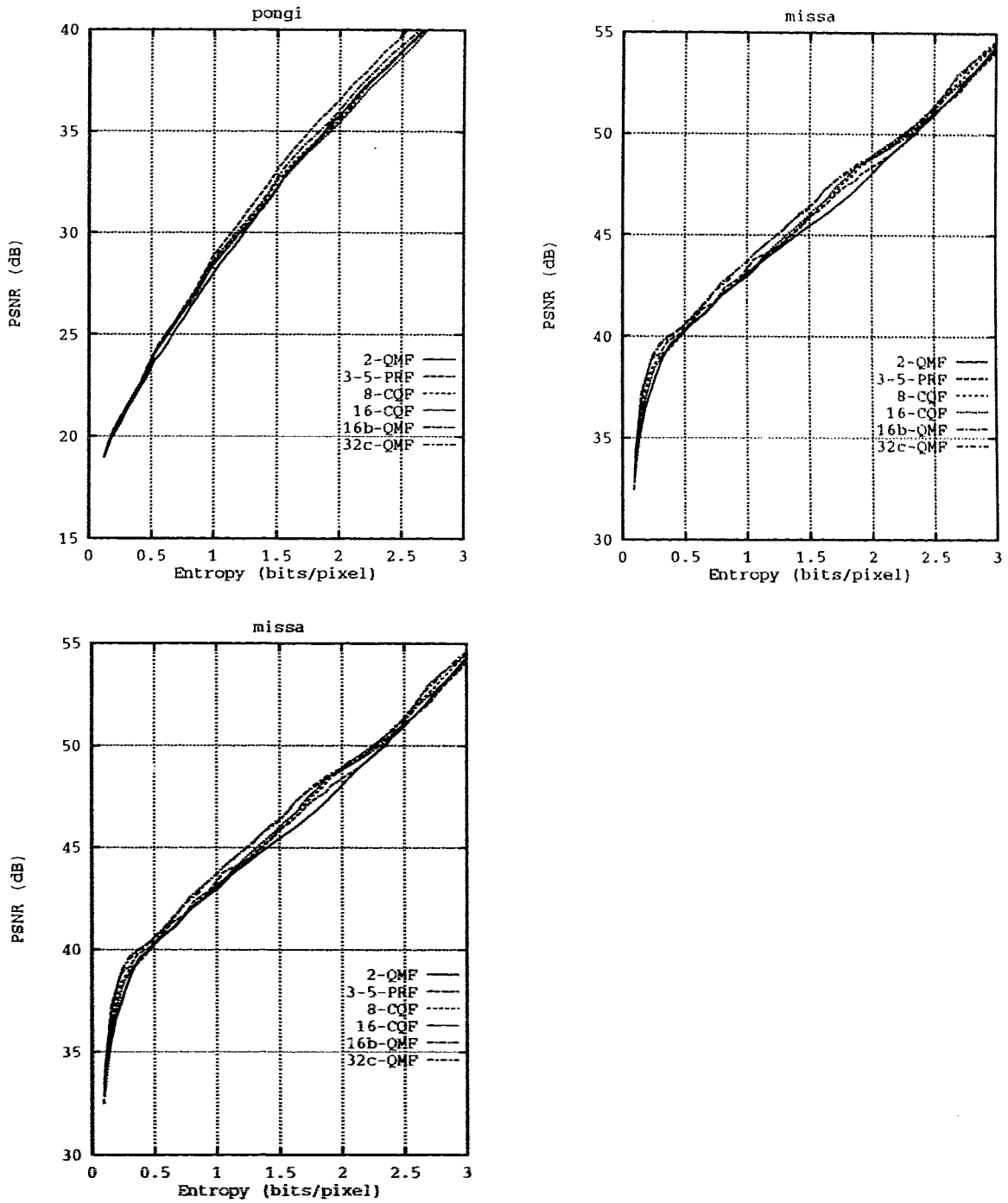


Figure 5.7: PSNR versus Entropy for the TS System with the 3-5-PRF Temporal Filter Set

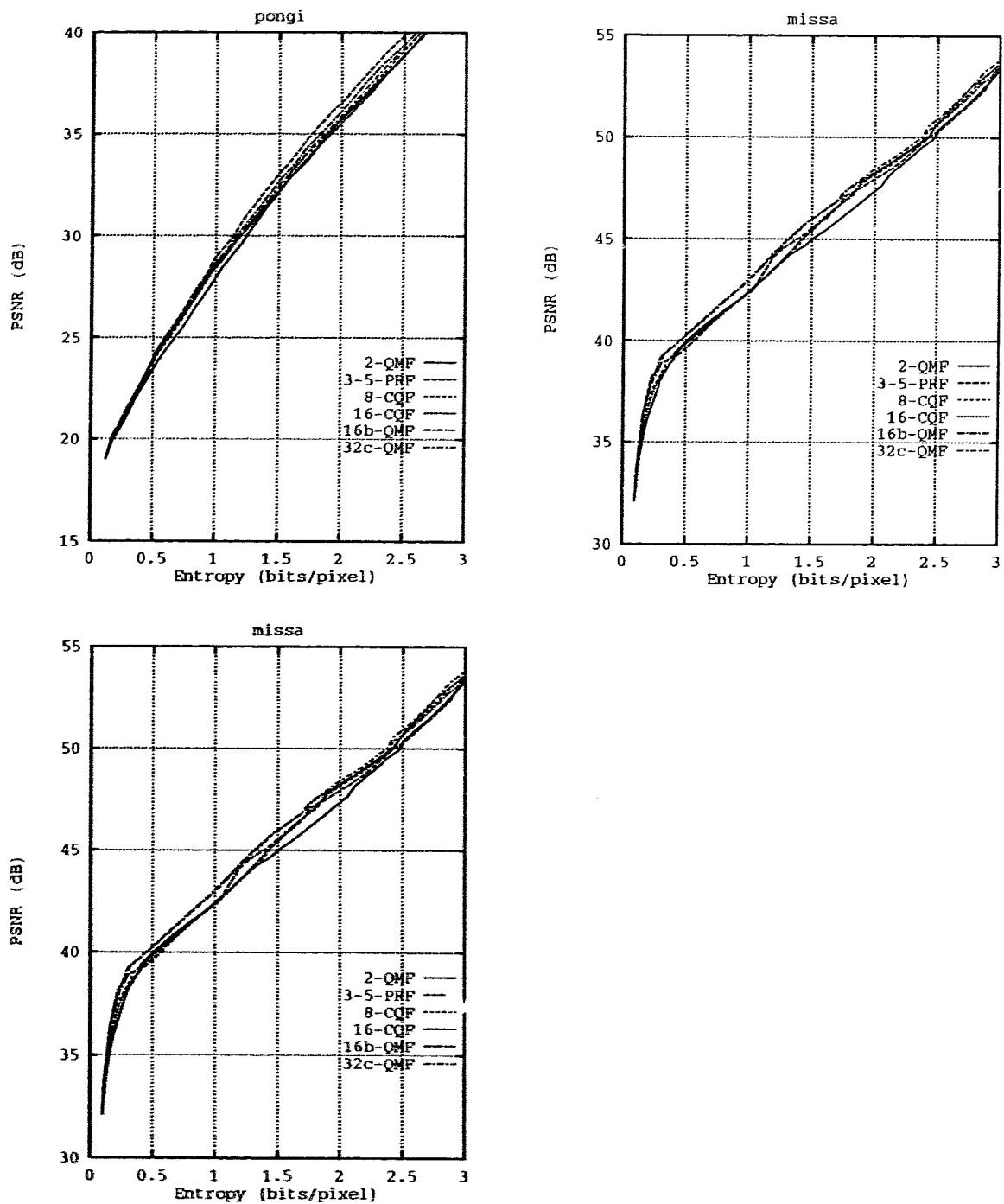


Figure 5.8: PSNR versus Entropy for the TS System with the 8-CQF Temporal Filter Set

pongi graph, the MS codec results show that the best filter set is shared by the 3-5-PRF and the 2-QMF filter sets. The MT codec results show that the short 2-QMF filter has the best performance and that the codec performance drops significantly as the filter length increases. The temporal filter set reconstruction delay causes this decrease in performance, because poorer block matches occur when the MC algorithm is forced to use a previous search frame a number of frames back in time compared to a frame closer to the present frame. In the *missa* graph, the MS system results show that the 8-CQF filter set is marginally better than the other filter sets; the MT system results are the same as in *pongi*. In the *sales* graph, the MS system results show that all the filter sets perform similarly at low rates and the 2-QMF filter set performs marginally better than the others at high rates; the MT results are the same as those found with the previous sequences. Compared to MT, MS coding performs better data compression, by as much as 3 dB in *pongi* and 5 dB in *sales*.

Given the results for the five systems, S, T, TS, MS, and MT, Figure 5.10 shows PSNR versus entropy performance comparisons between each system and the M and MDCT systems. The 2-QMF and 3-5-PRF filter sets were used for all temporal and spatial filtering banks respectively. For the *pongi* sequence, the system performances ranked from best to worst are M, MS, MDCT, MT, TS, S, and T. With high motion, the codecs using MC perform better than those using only subband filtering. The TS codec outperforms either S or T codecs alone. This is expected, because some of both the temporal and spatial redundancies are removed. For the *missa* sequence, the performance rankings from best to worst are TS, MDCT, S, T, MS, M, and MT. In *missa*, the performance of the TS codec is better than any codec using M. It appears that, for the low motion and low detailed background video sequences, the TS codec has advantages over the M codec. Because there is low motion and a low detailed background, the high frequency subbands contain little information. As a result, most of the information is compacted into the lowest frequency subbands, and so the TS codec performs well. For the *sales* sequence, the performance rankings from best to worst are MS, MDCT, TS, T, M, MT, and S. In this sequence, the medium motion and high detailed background benefits from M coding to remove temporal redundancies.

Table 5.4 records the weighted variances and correlations coefficients for encoded *pongi* and *missa* sequences. The variances are adjusted according to the filter weighting factors. One- and two-dimensional weighting factors are shown in Tables 5.1 and

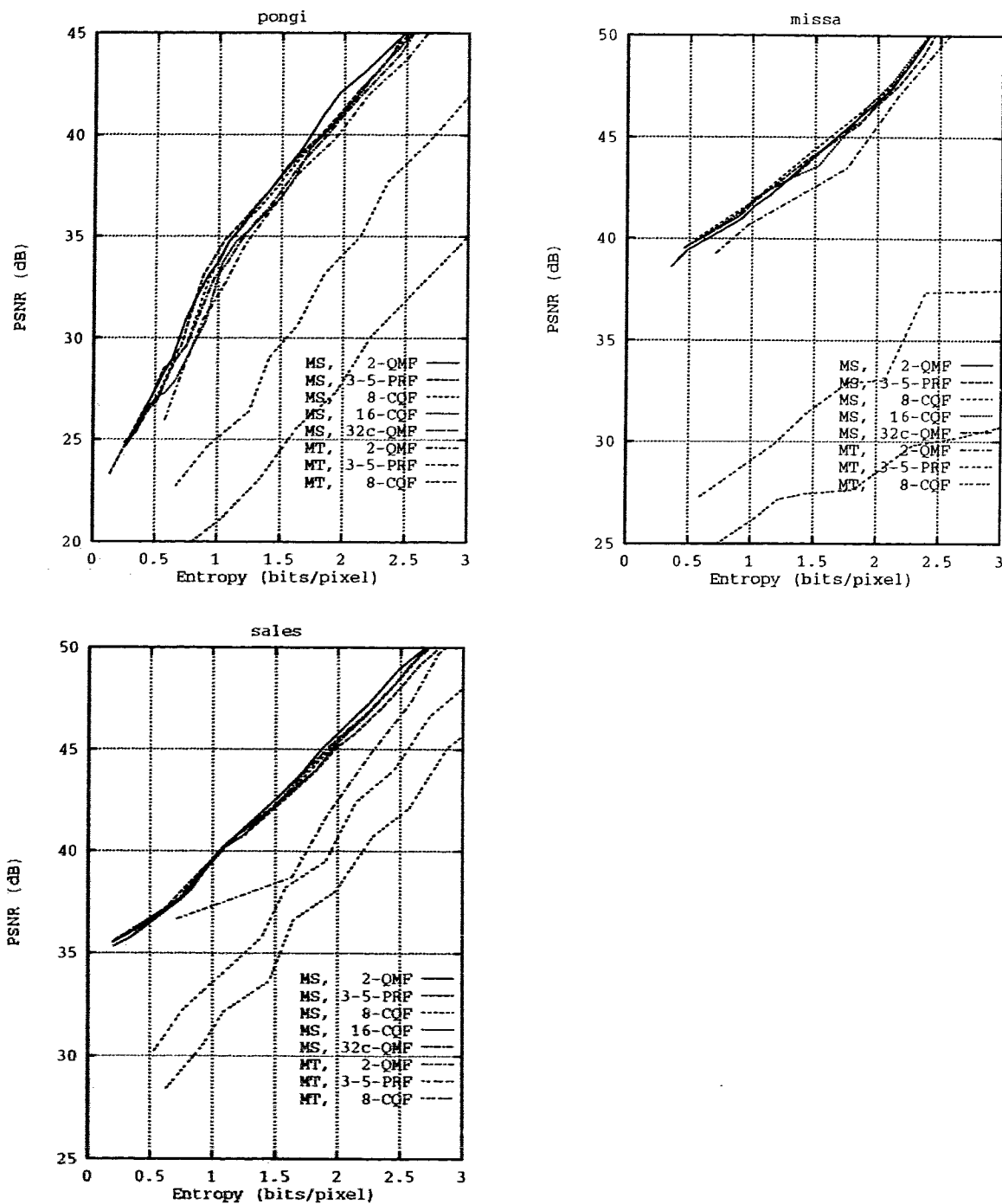


Figure 5.9: PSNR versus Entropy for the MS and MT Systems

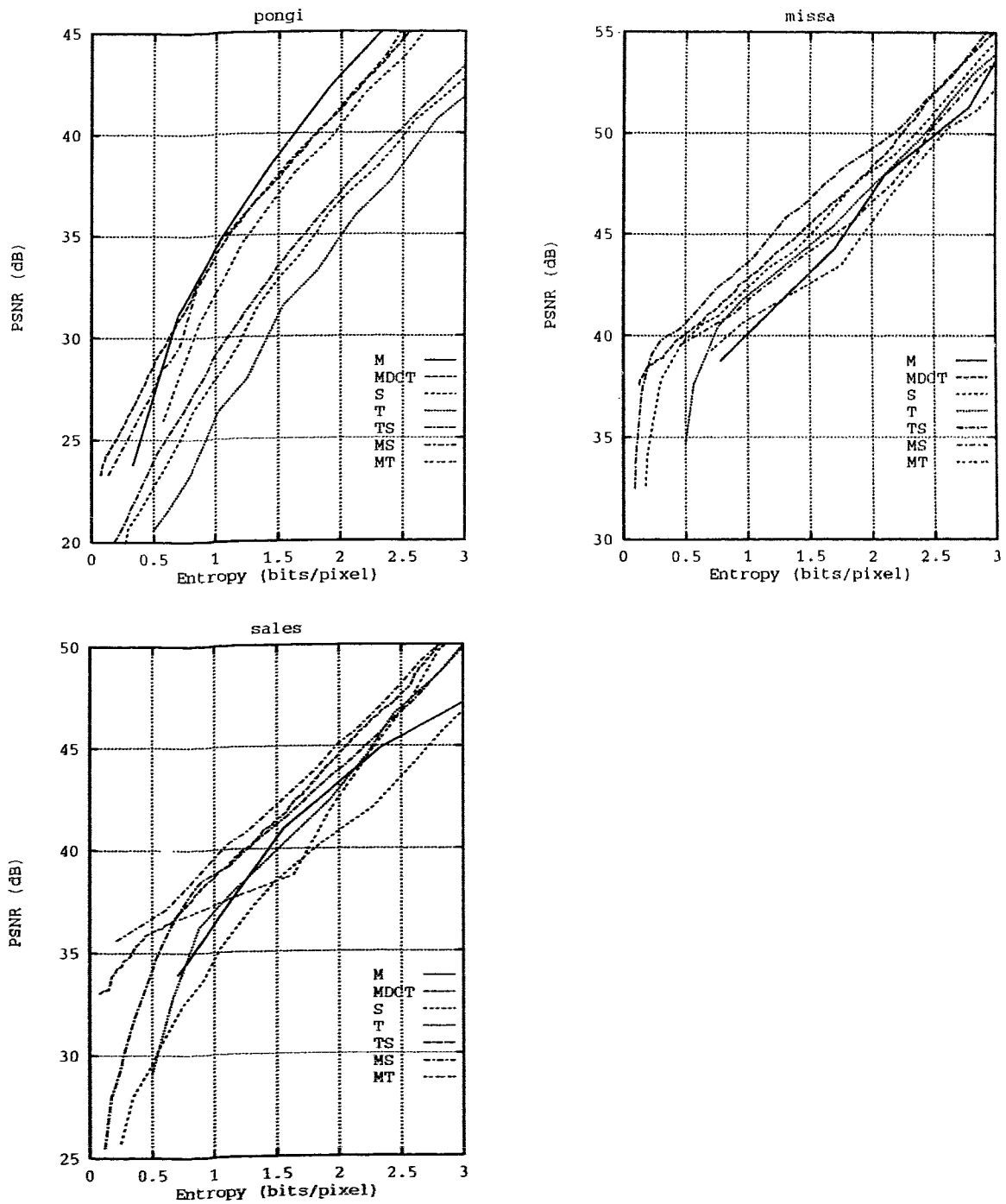


Figure 5.10: PSNR versus Entropy for Group 2 Systems: S, T, TS, MS and MT

Table 5.4: Group 2 Correlation and Weighted Variance Video Statistics

sequence	codec	$i = \text{band}$	$w_i \sigma_i^2$	ρ_x	ρ_y	ρ_z
pongi	S	ll 1	599.583	-0.1330	-0.0609	-0.0527
		lh 2	143.607	0.3720	-0.2956	0.4030
		hl 3	129.564	-0.2392	0.2475	-0.2444
		hh 4	27.351	-0.1956	-0.2598	-0.1982
pongi	T	l 1	186.030	-0.0848	0.0773	-0.1532
		h 2	367.391	0.2807	0.5946	-0.0611
pongi	TS	l-ll 1	249.933	0.0779	-0.0619	-0.2008
		l-lh 2	100.880	0.6829	-0.2822	0.4473
		l-hl 3	48.005	0.1638	0.2267	-0.1667
		l-hh 4	10.778	0.2290	-0.2477	-0.1240
		h-ll 5	200.098	-0.2769	0.2293	-0.0896
		h-lh 6	42.727	-0.3590	-0.3274	-0.1008
		h-hl 7	81.559	-0.4940	0.2548	0.0636
		h-hh 8	16.571	-0.4927	-0.2692	0.0690
pongi	MS	ll 1	49.256	-0.0808	0.0705	-0.0557
		lh 2	17.867	-0.0694	-0.1571	0.0109
		hl 3	35.722	-0.0405	0.1441	-0.0368
		hh 4	10.528	-0.0527	-0.1704	-0.0324
pongi	MT	l 1	76.097	0.2585	0.4630	0.0211
		h 2	63.219	0.1919	0.4479	-0.0112
missa	S	ll 1	16.397	0.0621	0.0902	0.8323
		lh 2	9.162	0.9010	-0.0635	0.9435
		hl 3	2.967	0.1563	0.3093	0.2675
		hh 4	1.909	0.7568	0.3148	-0.8043
missa	T	l 1	5.494	0.0323	0.2493	0.4597
		h 2	3.366	-0.2098	0.1666	0.3586
missa	TS	l-ll 1	14.959	0.0867	0.1088	0.7683
		l-lh 2	8.908	0.9259	-0.0616	0.9450
		l-hl 3	1.889	-0.0398	0.4223	0.5015
		l-hh 4	0.188	-0.1713	-0.1851	0.1850
		h-ll 5	0.956	0.2732	0.1089	0.0350
		h-lh 6	0.254	0.0091	-0.1167	0.0433
		h-hl 7	1.077	0.5000	0.1092	0.2828
		h-hh 8	1.721	0.8582	0.3696	0.9168
missa	MS	ll 1	2.404	0.1151	0.0779	0.0426
		lh 2	0.966	0.0828	-0.0169	0.0206
		hl 3	2.287	0.2756	0.0989	-0.1008
		hh 4	1.064	0.3654	0.1128	-0.3171
missa	MT	l 1	4.123	0.2228	0.4301	0.0258
		h 2	2.886	0.0315	0.3644	0.0490

5.2. The subband variances are largest for the low-pass filtered subbands, which shows how subband filtering can compact energy into a few subbands. The subbands ll , l and $l-ll$ for the S, T, and TS codecs respectively, have been DPCM encoded, so their variances have been significantly decreased, similar to the PCM to DPCM decrease in variance. The variances and pixel correlations for these bands would be similar to the PCM values if DPCM were not used here. The values for the ρ_x and ρ_y decrease to the range of 0.1–0.5 for subband filtering codecs, and to even lower values when MC is also used.

The computational load in terms of numbers of multiplications and additions for

this group of systems is calculated. The computational load for subband codecs is a function of the number of filter taps in the synthesis filters. If the same filters are used, the number of multiplications and additions per frame for an S and T system is identical:

$$\begin{aligned}\Gamma_m &= LWH && \text{per frame} \\ \Gamma_a &= (L-1)WH && \text{per frame}\end{aligned}, \quad (5.3)$$

where L is the number of filter taps. The computational load for a TS or ST system is twice that of an S or T system, since they are simply cascaded one after the other. The MS and MT codecs have a computational load equal to the sum of an M and S, or T, system respectively. If the 3-5-PRF and 2-QMF filter sets are used for the spatial and temporal filtering respectively, then the computational load measured in multiplications/additions for the S, T, TS, MS, and MT systems to encode an 8×8 pixel macro block is 256/192, 128/64, 640/256, 19041/36895, and 18913/36767 computations respectively. The computational load for subband filtering is higher than just the DCT, but is two orders of magnitude below the full search M codec load. For the *missa* sequence, the TS system, with a low computational load, performed better than any system that used MC. Proper selection of coding tools is therefore important to codec performance and computational load.

5.3.3 Video Codec Results for Group 3: SM, SM1, TM, and TM1

The next group of codec results to be presented is that consisting of the pair-wise SM, SM1, TM, and TM1 systems. Figures 5.11-5.14 show these results. As before, the performance of several filter sets was recorded.

The SM system performance is shown in Figure 5.11. In the figure, the curves cross each other from low to high rates. The slope of the performance curves tend to change at points where bits are first assigned to a subband for the first time. For example, consider the first five bit allocations for the 3-5-PRF filter set. The allocations are 2000, 2200, 2220, 3220, and 2222 bits, where abcd represents bit allocations to bands ll, lh, hl, and hh respectively. The entropy at these rates is 0.175, 0.332, 0.489, 0.594, and 0.790 bits/pixel. As the figure indicates, the 3-5-PRF curve changes slope twice in this bit rate interval corresponding with the third and fifth bit allocations where

the hl and hh subbands respectively are first assigned bits. For the *pongi* sequence, the best filter sets for the range tested were the 3-5-PRF and 8-CQF filter sets. The 3-5-PRF at high rates was a clear winner. For the *missa* sequence, the 8-CQF and 32c-QMF filter sets performed best at low rates while the 3-5-PRF filter set performed best at high rates. For the *sales* sequence, the filter sets performed best with the 32c-QMF followed in decreasing order by the 16-CQF, 8-CQF, 3-5-PRF, and 2-QMF filter sets. The performances of the codec when using the 3-5-PRF, 8-CQF, 16-CQF, and 32c-QMF filter sets were usually within at least 1 dB of each other at most rates and at most instances much closer. This result shows that the filter choice is up to the codec designer and that the shorter kernel 3-5-PRF filter set is a good choice.

The SM1 system is a modified SM system. Here only the Ω'_S subband is M coded. Figure 5.12 shows this system PSNR versus entropy relationship results. In the *pongi* sequence, the 3-5-PRF filter set clearly outperforms the other systems; in the *missa* sequence, the 3-5-PRF, 16-CQF, and 8-CQF filter sets perform similarly and best among the filters and, in the *sales* sequence, the 3-5-PRF and 32c-QMF filters sets performed comparably best. The hypothesis that the SM1 system might have a performance close to the SM system cannot be made. This conclusion can be seen when comparing the SM and SM1 performances in Figure 5.15 where the best Group 2 codecs performances are given. This conclusion contradict the one made by Paek, Kim, and Lee (1992), where they conclude a SM1 codec performs better than SM codec.

In addition to being used to code an S system output, MC can also be used to code a T codec output. Figure 5.13 shows the PSNR to entropy performance relationship for a TM codec. Similarly to the T system, the system performance degrades as the filter length increases. The 2-QMF filter set performance was best for all three video sequences. With the addition of MC to a T codec, the system's performance increased by 5, 2, and 7 dB for *pongi*, *missa* and *sales* respectively, compared with that of the T system alone.

The TM1 system performance is shown in Figure 5.14. As with the SM1 and SM codec performance rankings, the TM1 codec performance was below that of the TM system. The 2-QMF filter set performed best again. In the *pongi* and *sales* sequences, the TM1 performance curve ranges approximately half-way between the TM and T systems. But in the *missa* sequence, the TM1 performance degrades below that of

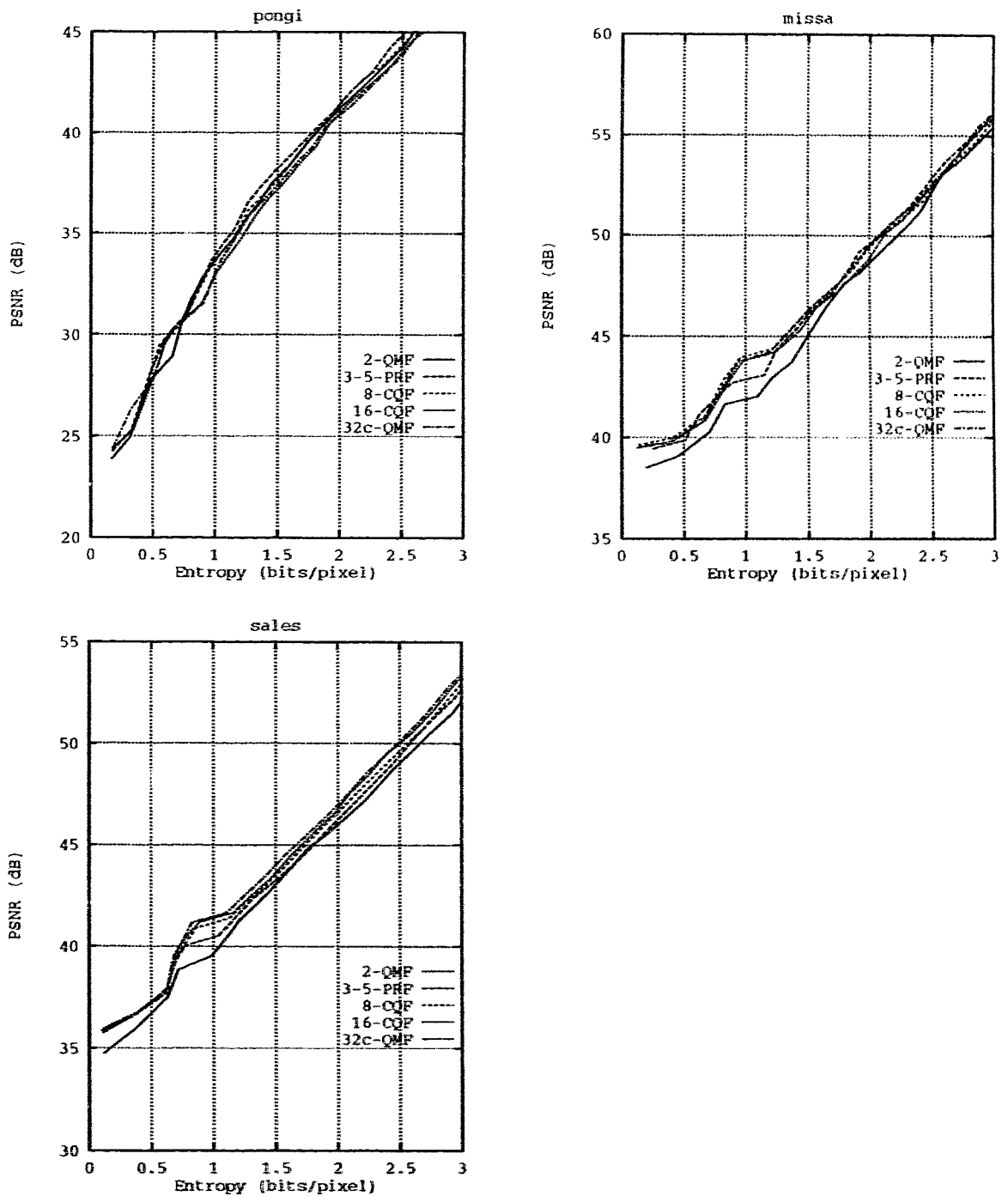


Figure 5.11: PSNR versus Entropy for the SM System

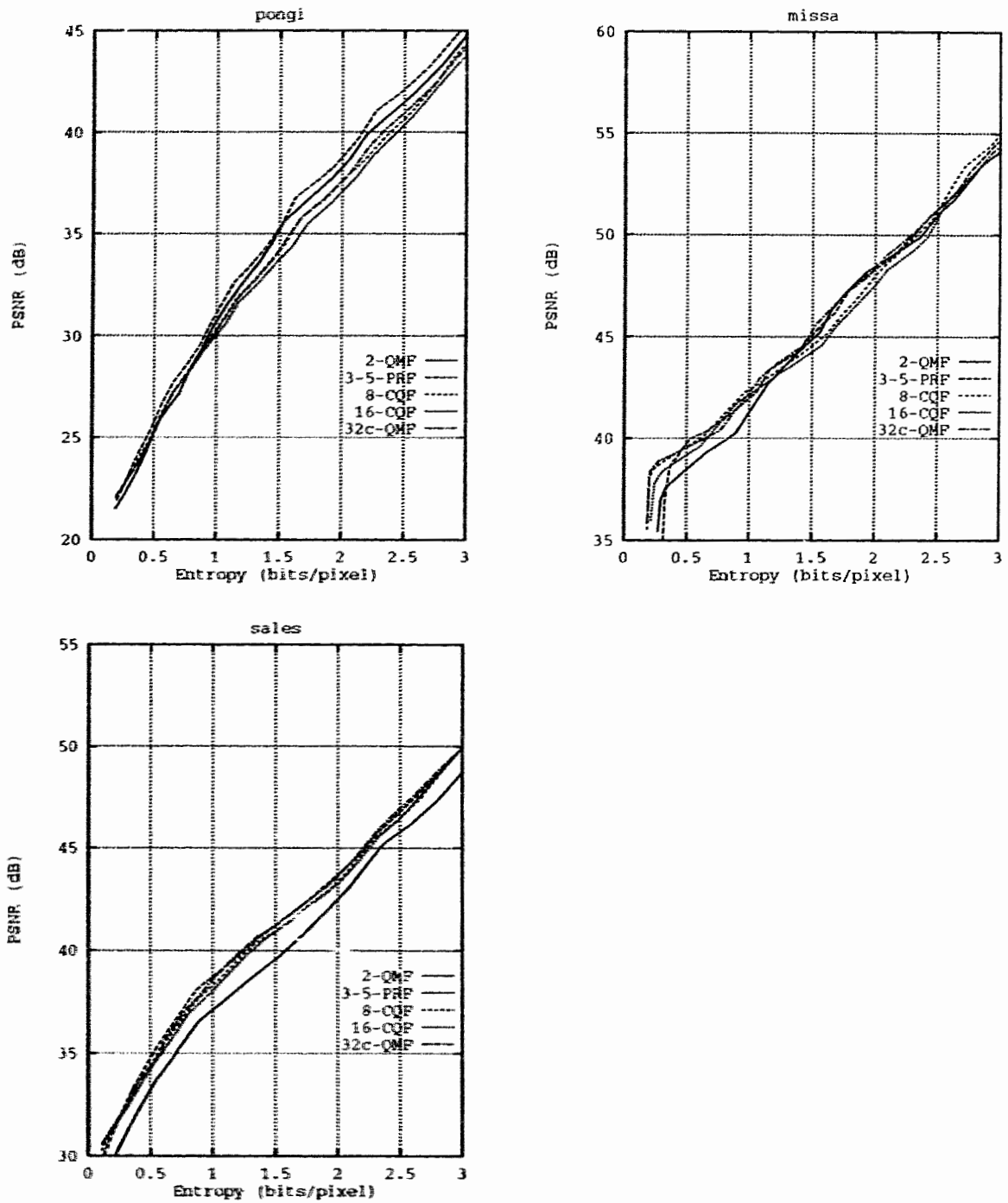


Figure 5.12: PSNR versus Entropy for the SM1 System

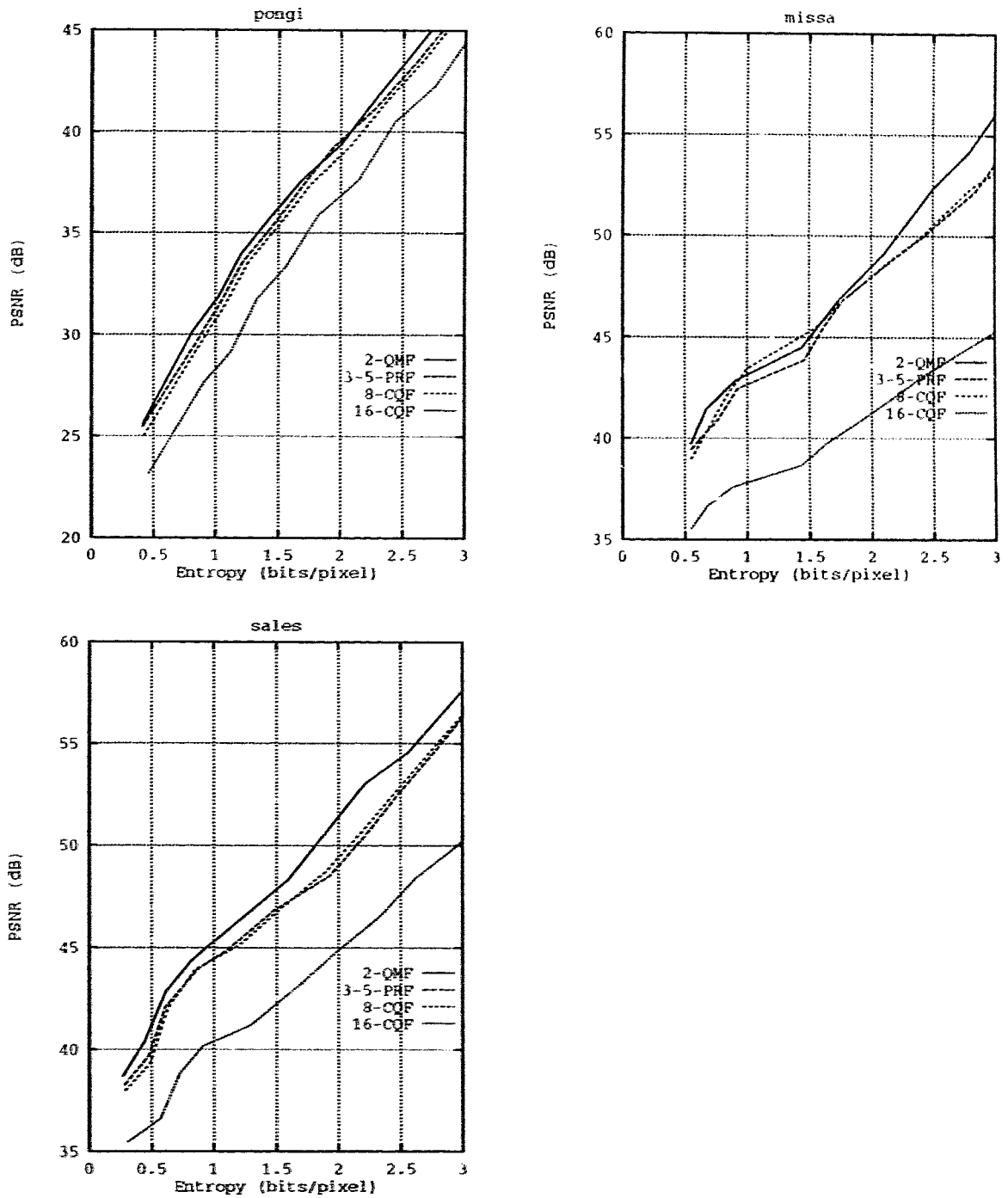


Figure 5.13: PSNR versus Entropy for the TM System

the T system. These results indicate there are benefits to M encoding the high-pass temporal subband.

A comparison of the best codec performances from the SM, SM1, TM, and TM1 systems is shown in Figure 5.15. The M and MDCT performances are also plotted. For the *pongi* sequence, the codec performance rankings are easily seen. From best to worst they are M, SM – MDCT, TM, SM1, and TM1 (a “–” between two labels implies equal or similar performance). The SM and MDCT performances cross each other with the SM system performing better at high rates, and the MDCT system performing better at lower rates. For the *missa* sequence, the performance rankings are SM, TM – MDCT, SMT, TM1, and M. SM performs best and, as seen in the Group 2 summary figure, the M system is outperformed by systems that use subband filtering. These results show the strengths of the SM system. The work by Paek, Kim, and Lee (1992) studied SM1-like systems and found them to perform marginally better than an MDCT codec. They compared the systems only at one rate, but they did show that blocking effects are reduced when spatial subband filtering is performed. The results here agree with this study in that there is merit to integrating subband filtering with MC especially since a multiresolution system is now possible. For the *sales* sequence, the performance rankings from best to worst are TM, TM1, SM, MDCT, SM1, and M. Here, the results show that codecs using temporal filtering and MC outperform the others and implies the presence of redundant temporal information. Similar temporal redundancies were found in this sequence by Gothe and Vaisey (1993), when multiple temporal search frames were used in an M coder.

Table 5.5 records the weighted variances and correlations coefficients after encoding the *pongi* and *missa* sequences. Again, the variances are altered by the filter weighting factors. When the variance and correlation statistics in this table are compared to the S and T results in Table 5.4, the values are significantly lower. The addition of MC has reduced the energy in the subband and is a decorrelation process.

Using the computational load formulas derived for each coding tool earlier, the load can be calculated for Group 3 systems. The SM codec computational load is lower than the MS codec, because the search size variable in the decimated subband frames was halved to $p = 4$, while the block size remained constant. The search variable is halved to keep the search region in the S codec subband frames, which are decimated by two in each spatial direction, the same as in the full sized frames. The smaller search

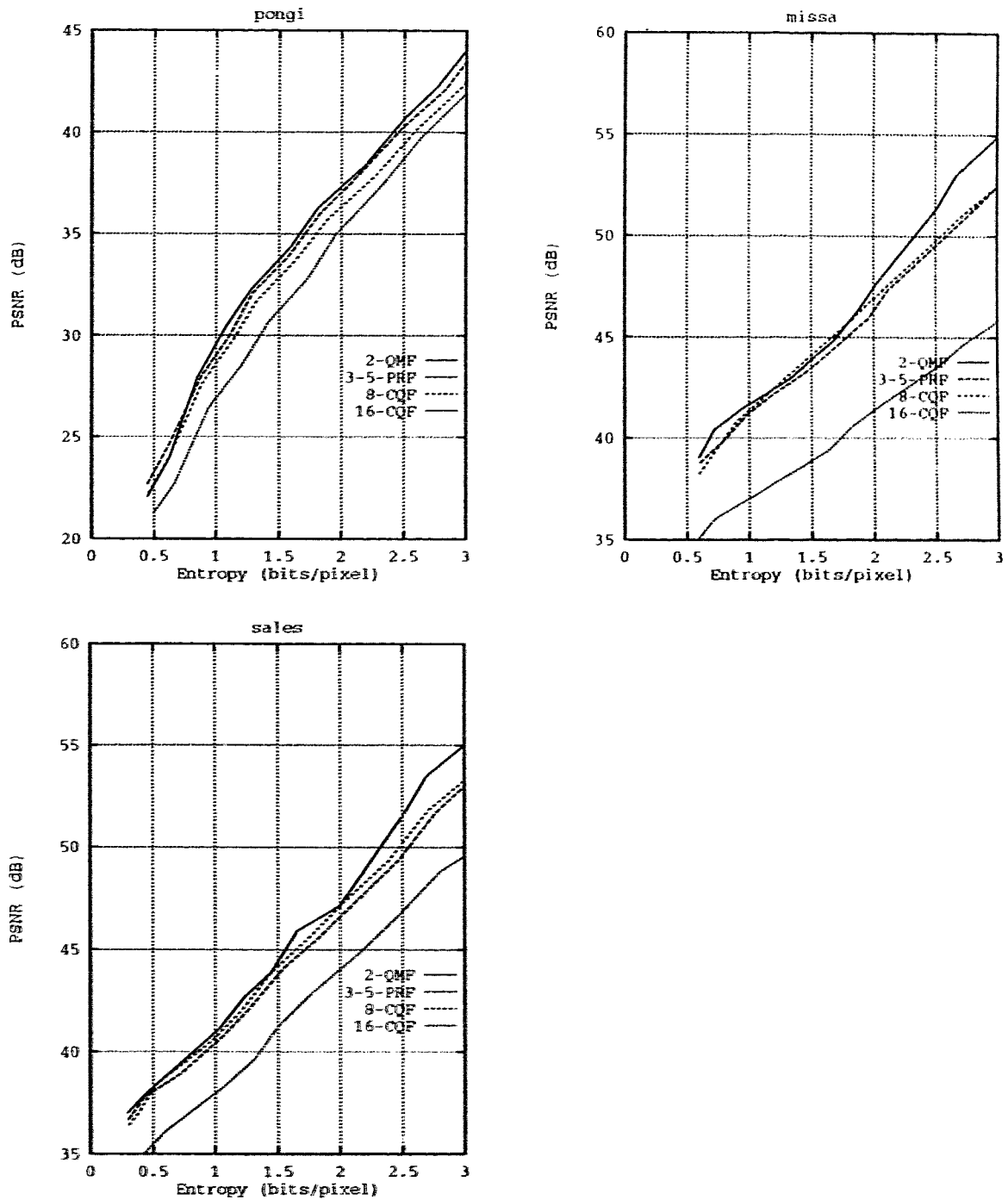


Figure 5.14: PSNR versus Entropy for the TM1 System

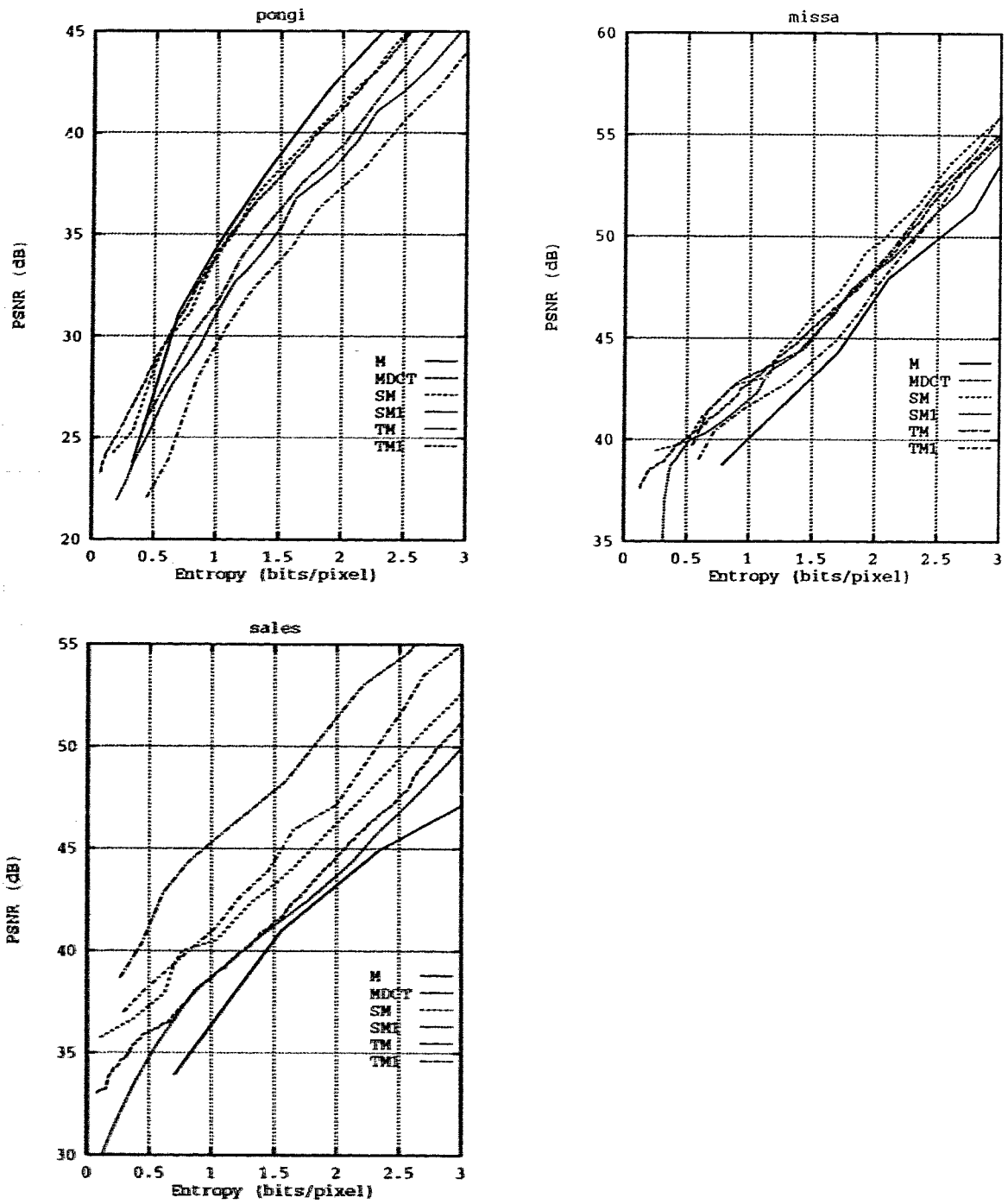


Figure 5.15: PSNR versus Entropy for Group 3 Systems: SM, SM1, TM, and TM1

Table 5.5: Group 3 Correlation and Weighted Variance Video Statistics

sequence	codec	i=band	$w_i \sigma_i^2$	ρ_x	ρ_y	ρ_z
<i>pongi</i>	SM	ll 1	98.291	0.0094	0.1444	-0.0193
		lh 2	19.846	-0.0540	-0.2775	-0.0264
		hl 3	43.699	-0.1415	0.2216	-0.0762
		hh 4	8.735	-0.1066	-0.2222	-0.0524
<i>pongi</i>	TM	l 1	94.482	0.3401	0.4747	0.0166
		h 2	84.280	0.1855	0.4850	-0.0174
<i>missa</i>	SM	ll 1	3.409	0.2560	0.1172	0.1119
		lh 2	0.705	0.0615	-0.1315	-0.0387
		hl 3	2.284	0.2632	0.1877	-0.0752
		hh 4	0.531	0.0705	-0.0509	-0.1176
<i>missa</i>	TM	l 1	5.042	0.3608	0.4740	0.0242
		h 2	2.069	0.0102	0.3803	-0.0053

window implies a lower computational load. The SM, SM1, TM, and TM1 system encoder computational complexity, measured in multiplications/additions per 8×8 macro block, is 5521/10479, 1572/2764, 18913/36767, and 9521/18415 respectively. Again, 3-5-PRF and 2-QMF filter sets were assumed for the spatial and temporal filtering operations.

5.3.4 Video Codec Results for Group 4: TSM, TSM1, MTS, SMT, and TMS

Results for the last codec groupings, the TSM, TSM1, MTS, SMT, and TMS systems, are presented here and discussed. Relying on the best filter set results from above and, if not labeled otherwise, the temporal and spatial filter sets used in these simulations were the 2-QMF and 3-5-PRF filter sets respectively. The PSNR to entropy relationship for each system is shown in Figure 5.16. For *pongi*, the figure shows that the M and MDCT codecs perform 2 to 3 dB better than any other system. Below these systems, the MTS, TMS, and SMT codecs have similar performances. For *missa*, the TSM, SMT, and TSM1 codecs perform best at low rates, and the TSM performance increases by 1.5 dB over the others at high rates. The TMS, MDCT, M and MTS systems perform up to 5 dB worse than these systems. For *sales*, the TMS codec performance is significantly higher than that of the other systems. Only the TSM codecs performance gets close at very low and high rates. The remaining rankings in descending order are SMT, TSM1, MDCT, and M. Note: an anomaly where the curves for the SMT and MTS codecs decrease at 0.5 bits/pixel as the bit rate increases in the *sales* sequence is inexplicable, because the BFOS bit allocation

algorithm is supposed to select ever increasing performance allocations as the bit rate increases. In the high motion *pongi* sequence, the M and MDCT codec performances dominate and, in the medium to low motion *sales* and *missa* sequences, the subband filtering systems cascaded with MC perform well.

Table 5.6 records the weighted variances and correlations coefficients for the encoded *pongi* and *missa* sequences.

The computational load for these triple coding tool video codec configurations is the largest. The Group 4 computational load estimated in multiplications/additions per 8×8 pixel macro block is 5649/10543 computations for the SMT and TSM encoders, 1042/1542 computations for the TSM1 encoder, and 19169/36959 computations for the TMS and MTS encoders. The TSM1 complexity is lower than that of the other systems in Group 4, because only the lowest frequency subband sequence is M coded, resulting in approximately one-eighth the computations.

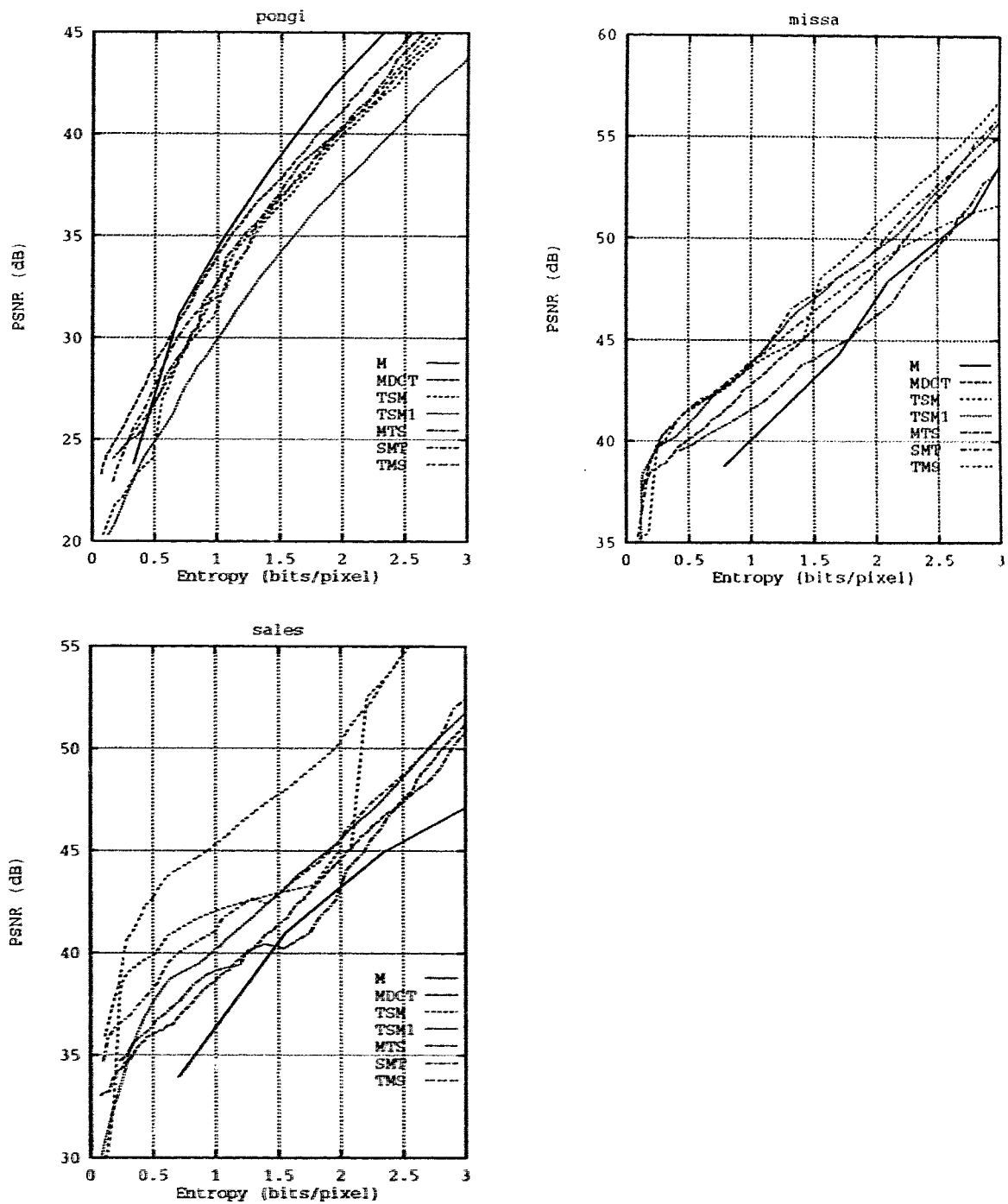


Figure 5.16: PSNR versus Entropy for Group 4 Systems: TSM, TSM1, MTS, SMT, and TMS

Table 5.6: Group 4 Correlation and Weighted Variance Video Statistics

sequence	codec	$i = \text{band}$	$w_i \sigma_i^2$	ρ_x	ρ_y	ρ_z
pongi	TSM	l-l 1	90.665	0.2874	0.2421	-0.0614
		l-lh 2	14.274	0.1271	-0.2803	-0.0345
		l-hl 3	24.329	0.0666	0.2246	-0.0811
		l-hh 4	4.922	0.1197	-0.2238	-0.0539
		h-l 5	69.902	-0.1771	0.1576	-0.0202
		h-lh 6	14.255	-0.2371	-0.2823	0.0050
		h-hl 7	38.647	-0.3199	0.2329	-0.0302
		h-hh 8	7.720	-0.2969	-0.2295	-0.0305
pongi	SMT	ll-l 1	70.549	0.2130	0.2124	0.0156
		ll-h 2	64.649	-0.1690	0.1440	0.0074
		lh-l 3	12.303	0.1090	-0.2730	-0.0102
		lh-h 4	12.751	-0.2393	-0.2804	0.0324
		hl-l 5	25.818	0.0405	0.1993	-0.0582
		hl-h 6	27.879	-0.3352	0.2438	0.0138
		hh-l 7	5.470	0.0917	-0.2187	-0.0385
		hh-h 8	5.395	-0.3180	-0.2404	-0.0217
pongi	TMS	l-l 1	43.909	0.1594	0.1244	0.0147
		l-lh 2	14.170	0.1191	-0.1823	0.0319
		l-hl 3	23.603	0.1279	0.1397	-0.0046
		l-hh 4	6.664	0.1083	-0.1801	0.0018
		h-l 5	6.213	-0.2098	-0.2038	-0.0036
		h-lh 6	24.954	-0.1897	0.1655	-0.0357
		h-hl 7	24.954	-0.1897	0.1655	-0.0357
		h-hh 8	6.780	-0.2177	-0.1917	-0.0311
pongi	MTS	l-l 1	33.121	0.0737	0.1094	0.0348
		l-lh 2	11.100	0.0364	-0.1556	0.0450
		l-hl 3	22.524	0.0647	0.1359	-0.0131
		l-hh 4	6.540	0.0468	-0.1817	-0.0180
		h-l 5	29.165	-0.1641	0.0669	-0.0080
		h-lh 6	9.895	-0.1372	-0.1695	-0.0006
		h-hl 7	17.976	-0.1272	0.1503	-0.0095
		h-hh 8	5.307	-0.1572	-0.1738	-0.0006
missa	TSM	ll-l 1	4.733	0.4042	0.2310	-0.0151
		ll-h 2	0.593	0.2597	-0.1977	0.0005
		lh-l 3	1.201	0.1207	0.2528	0.0034
		lh-h 4	0.169	-0.1220	-0.1585	0.0210
		hl-l 5	0.789	0.1934	0.0506	-0.0012
		hl-h 6	0.232	-0.0023	-0.0947	-0.0223
		hh-l 7	0.892	0.3398	0.1476	0.0094
		hh-h 8	0.227	0.0058	-0.0723	-0.1078
missa	SMT	ll-l 1	3.072	0.3384	0.1689	0.0157
		ll-h 2	1.458	0.2911	0.1134	0.0126
		lh-l 3	0.516	0.2047	-0.1706	-0.0457
		lh-h 4	0.263	-0.0233	-0.0832	0.0799
		hl-l 5	1.267	0.1595	0.2418	0.0489
		hl-h 6	0.969	0.3363	0.1349	0.0865
		hh-l 7	0.236	-0.0202	-0.0975	0.0316
		hh-h 8	0.250	0.0704	-0.0424	0.1169
missa	TMS	l-l 1	2.269	0.2752	0.1352	0.0650
		l-lh 2	0.791	0.3000	-0.0112	0.0756
		l-hl 3	1.252	0.1933	0.1884	-0.0901
		l-hh 4	0.296	-0.0257	-0.0849	-0.0485
		h-l 5	0.149	-0.0349	-0.0512	-0.0239
		h-lh 6	0.790	0.3021	0.0876	-0.0071
		h-hl 7	0.790	0.3021	0.0876	-0.0071
		h-hh 8	0.345	0.2930	0.0893	0.1927
missa	MTS	l-l 1	1.677	0.2051	0.1094	0.0529
		l-lh 2	0.635	0.2225	-0.0223	0.0344
		l-hl 3	1.227	0.2305	0.1485	-0.0450
		l-hh 4	0.428	0.2381	0.0486	0.0152
		h-l 5	1.016	0.0859	0.0667	0.0285
		h-lh 6	0.421	0.0405	-0.0222	0.0136
		h-hl 7	1.061	0.3044	0.0831	0.0166
		h-hh 8	0.519	0.3853	0.1239	0.2961

5.3.5 Video Codec Results Comparison of All Systems

The following discussion evaluates the performance of the best systems after all the systems are ranked one against the other. From the summary plots in each of the four groups, codec performance rankings were performed. Figures 5.17-5.19 show the rankings for the *pongi*, *missa*, and *sales* sequences respectively. The rankings were made by using a subjective evaluation of the performance curves above a bit rate of 1 bit/pixel. In the figures, the graph key represents codec rankings from best to worst. For all spatial and temporal subband filtering, the 3-5-PRF and 2-QMF filter sets were used respectively. In *pongi*, the M, SM, MS, MDCT systems perform best, with M leading at high rates, and MDCT leading at low. M-based codecs perform best for this high motion video sequence. In *missa*, the triplet systems, TSM, SMT, and TSM1 perform best, although the TS codec performance is sometimes within 0.5 dB. The results here show that three-dimensional subband filtering does not always perform poorer than M codecs. In *sales*, the TMS and TM systems perform nearly 5 dB higher than the next best systems, TSM and TM1. Now, considering the best performing codec results, Figure 5.20 shows the best top eight codecs for each video sequence. If the objective is to design a codec for high motion video that has reasonable performance for low motion video, the SM codec is a good candidate, since it ranks second, fifth, and fourth in *pongi*, *missa*, and *sales* respectively. Similarly, in the design of a codec for low and medium motion video, a TSM system performs well; in addition, the *pongi* results show it has reasonable performance for high motion video. The results show that the standard MDCT type codec performs comparably as well as a SM system for high motion video where it may be used, but it performs poorly for low motion video and its use is not suggested here.

In addition to evaluating codec performances, multiresolution codec capabilities are of interest when implementing a multi-tiered quality video service. Codecs that place M last and use subband filtering are easy to implement in a multiresolution video service. Conversely, if subband filtering is inside the MC feedback loop, all the subbands must be transmitted and used in the decoder, because the encoder reconstructs the search frames using all the subband information. These arguments also point to the benefits of the SM and TSM codec configurations. To present this more clearly, Figures 5.21-5.23 show block diagrams of the MDCT, SM, and

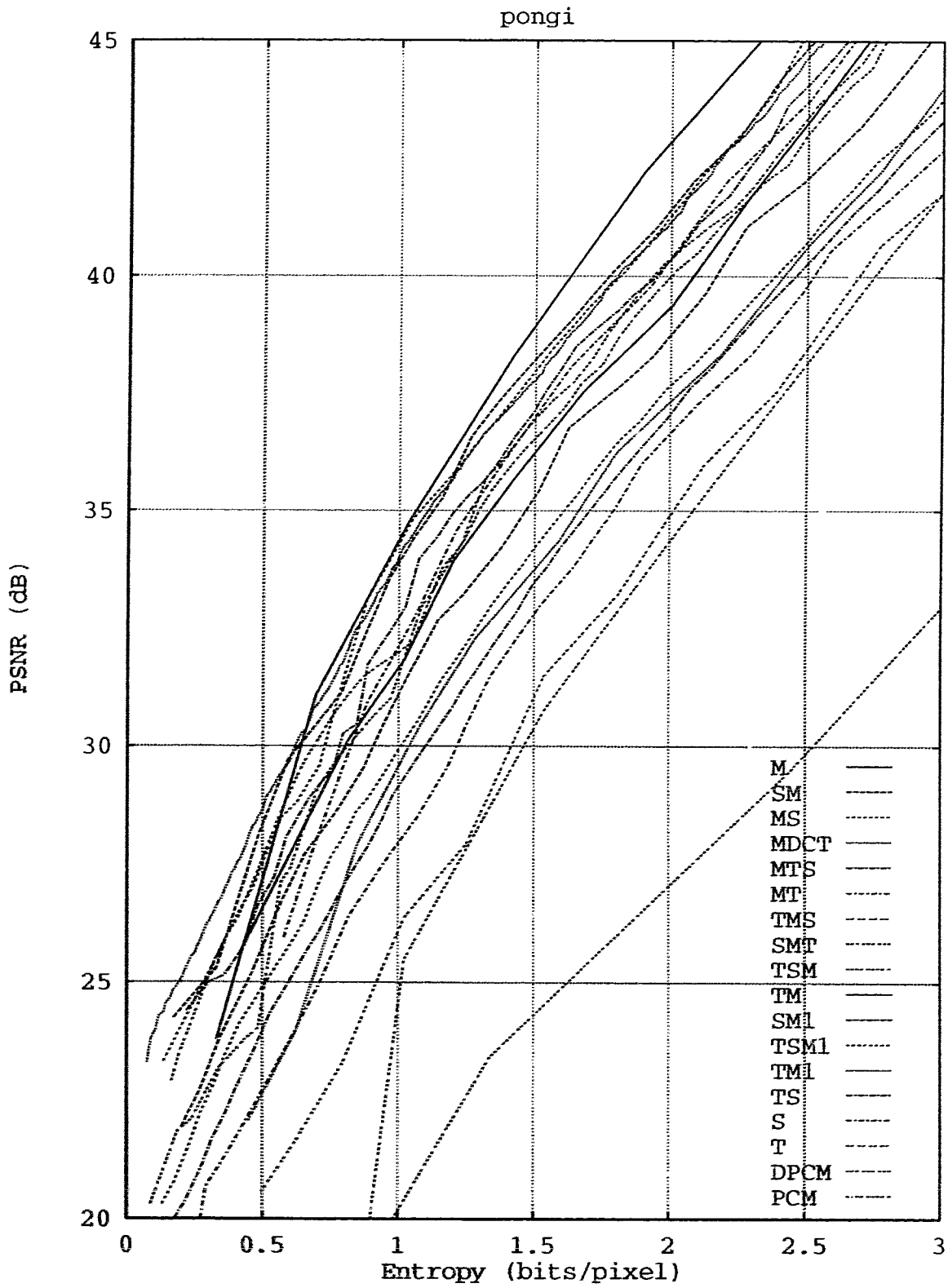
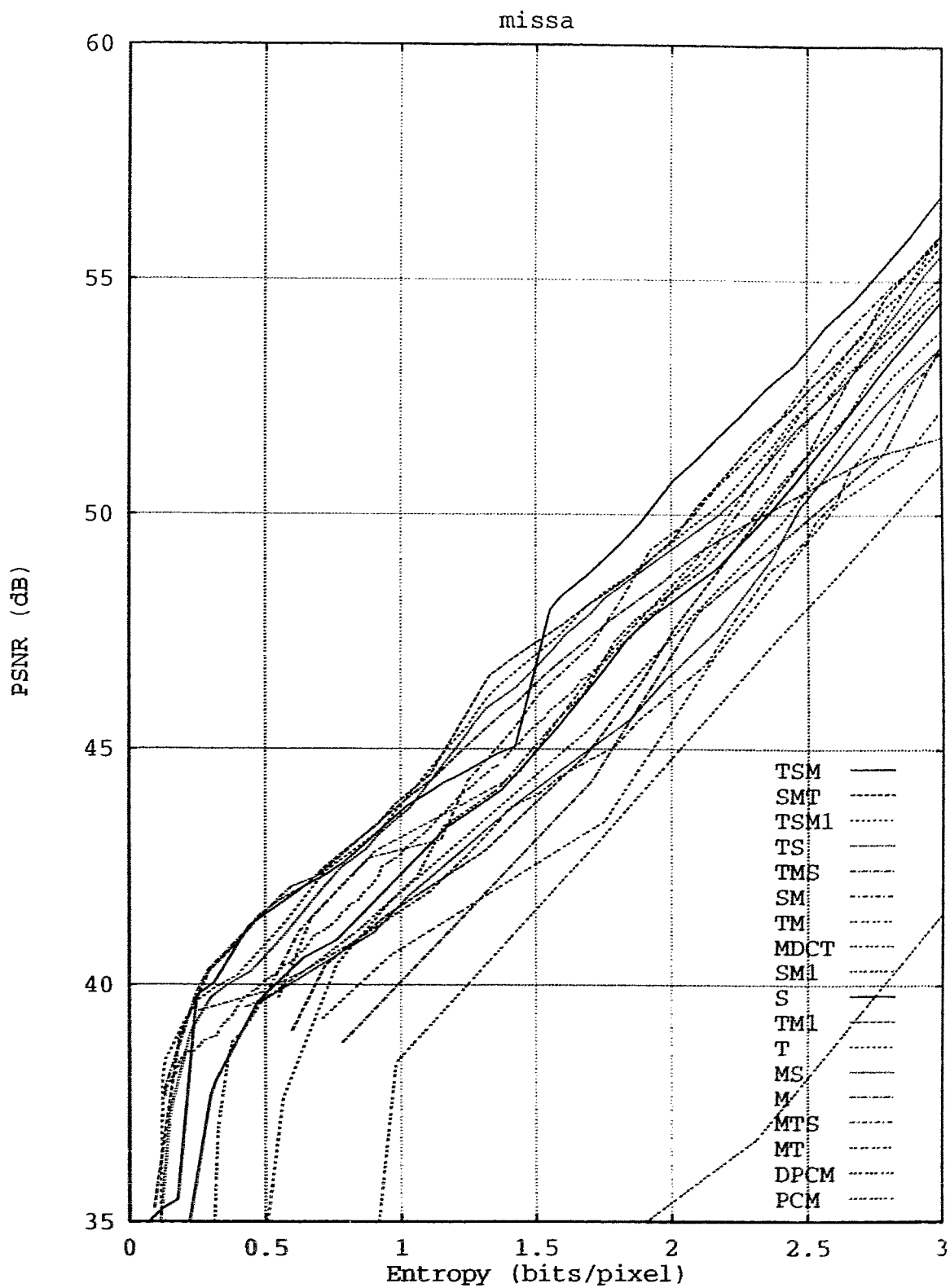


Figure 5.17: PSNR versus Entropy for All Systems, *pongi* Sequence

Figure 5.18: PSNR versus Entropy for All Systems, *missa* Sequence

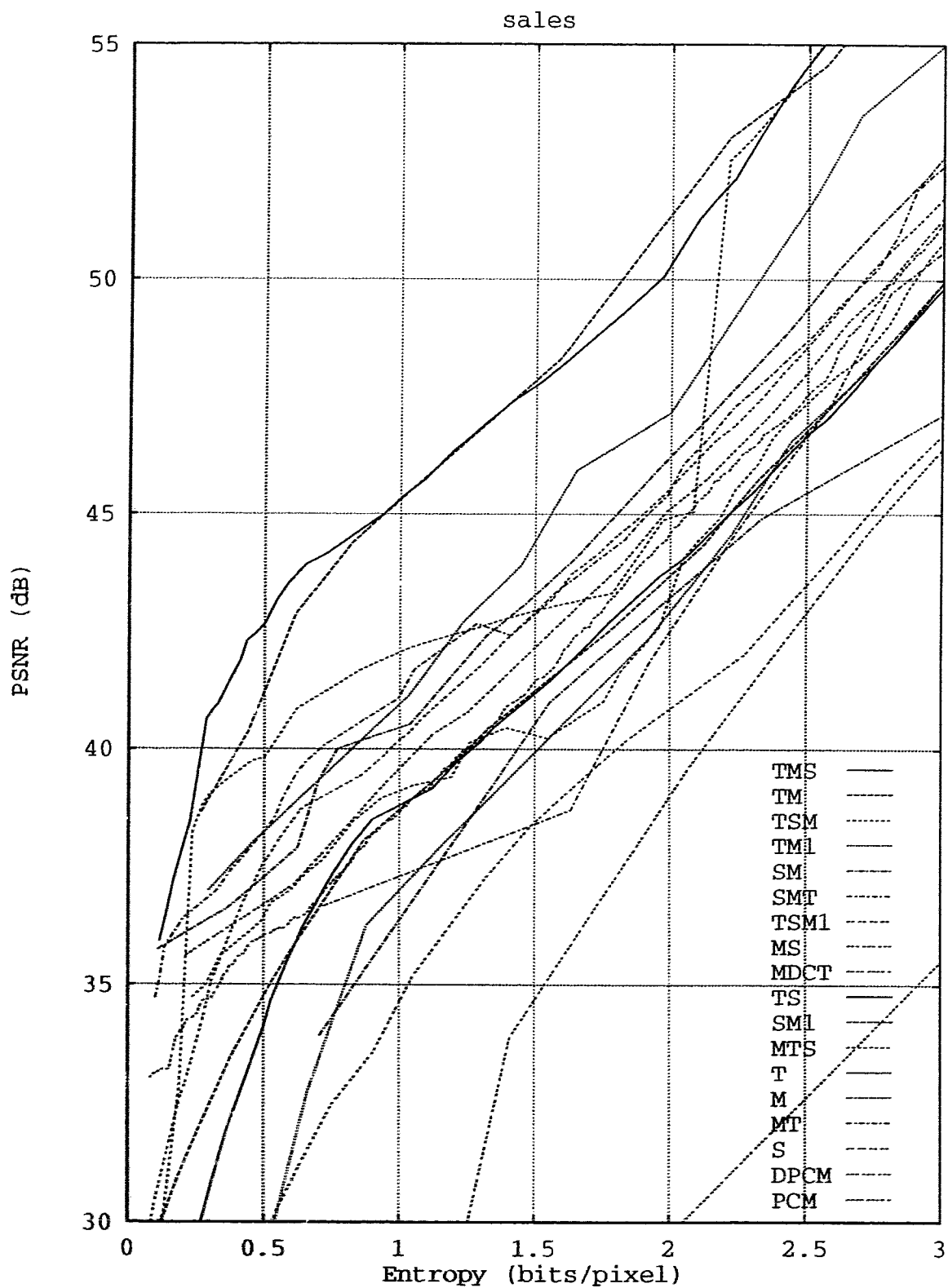


Figure 5.19: PSNR versus Entropy for All Systems, sales Sequence

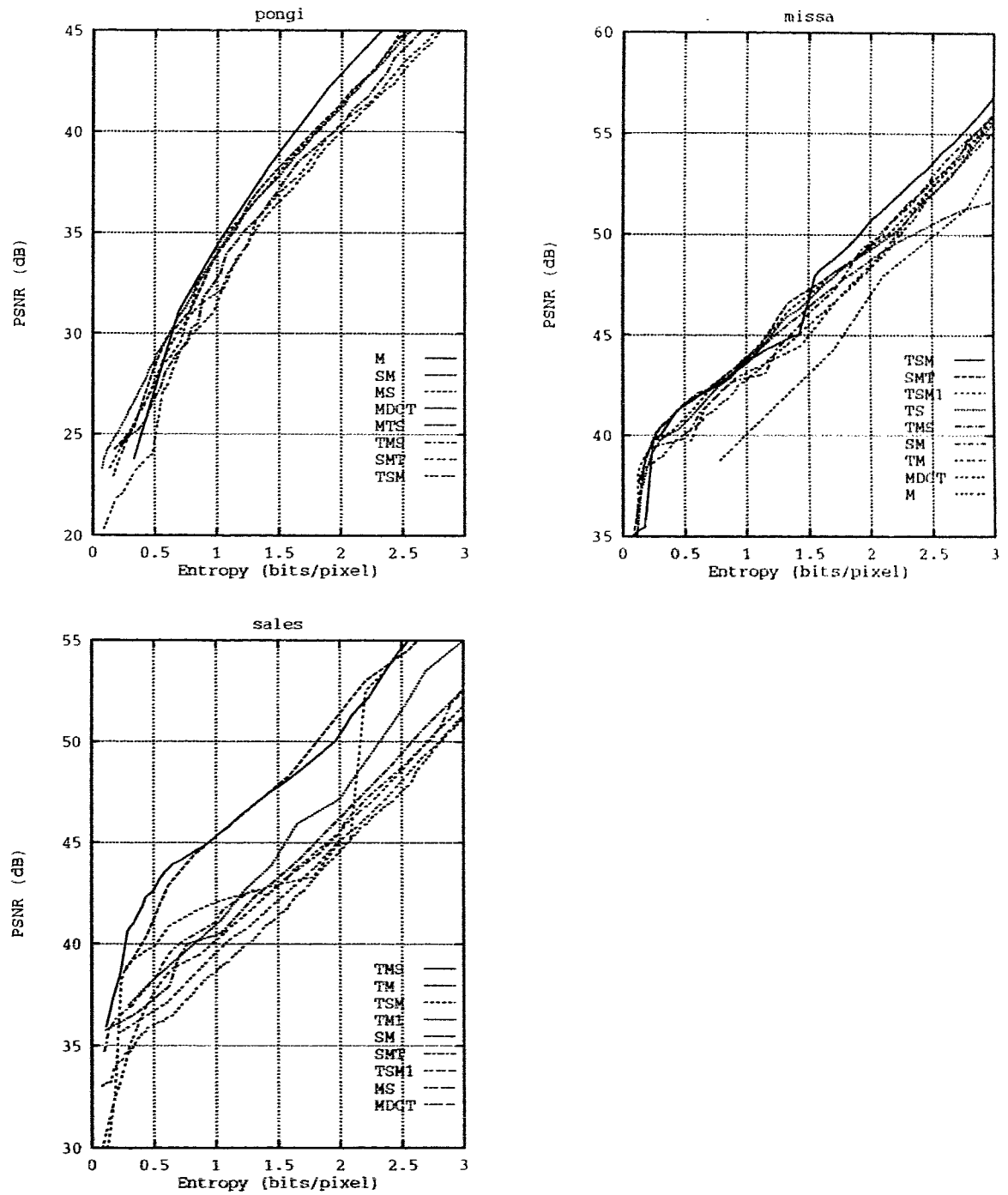


Figure 5.20: PSNR versus Entropy for the Best Performing Systems

TSM codecs. In the MDCT codec diagram the DCT is configured inside the MC feedback loop, whereas, in the the SM and TSM codec diagrams, the subband filtering operations occur before MC. The MDCT codec configuration is similar to the MC-subband filtering codecs that place MC before subband filtering, i.e., just replace the DCT with a subband filtering codec.

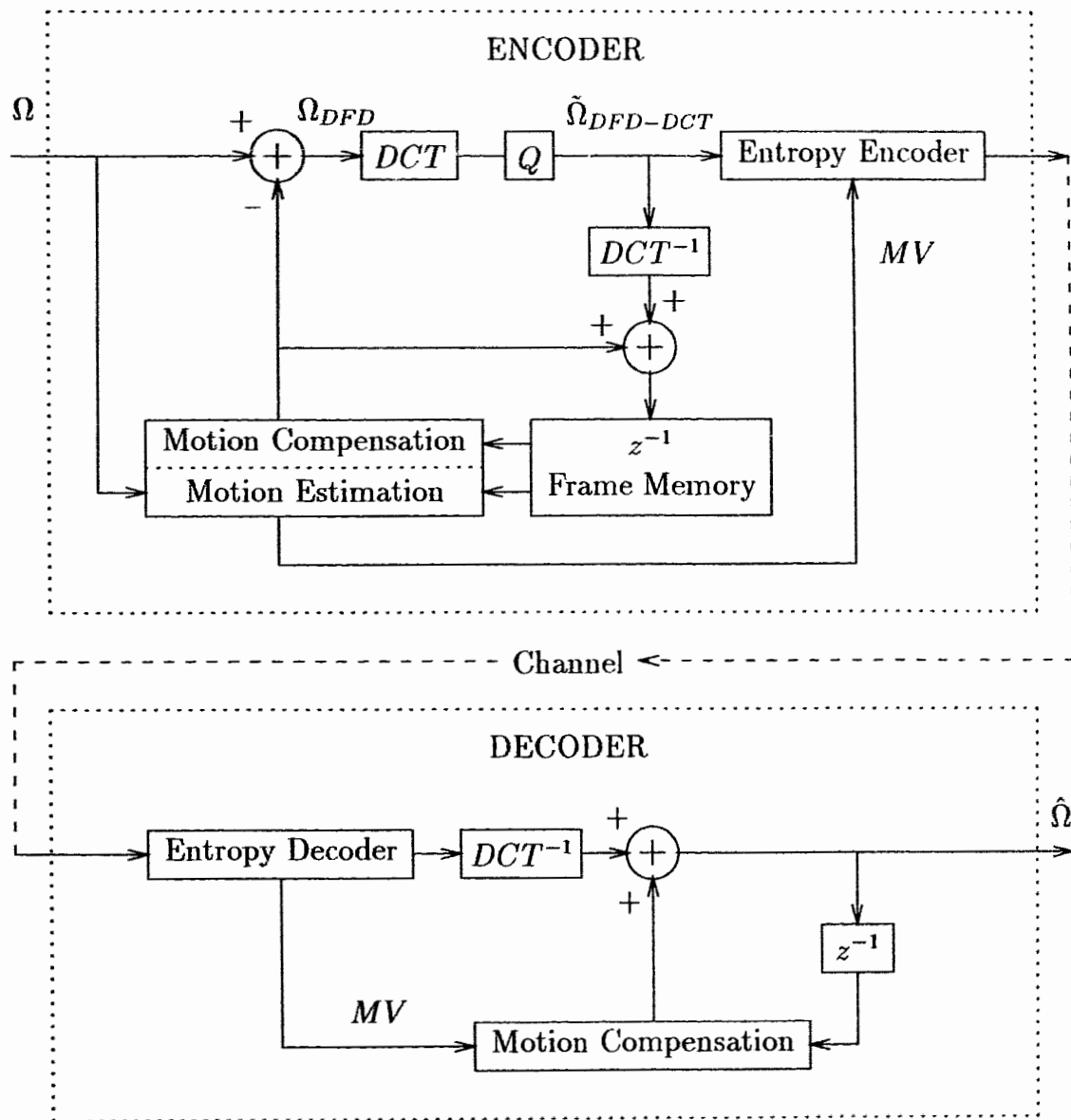


Figure 5.21: An MDCT Codec Block Diagram

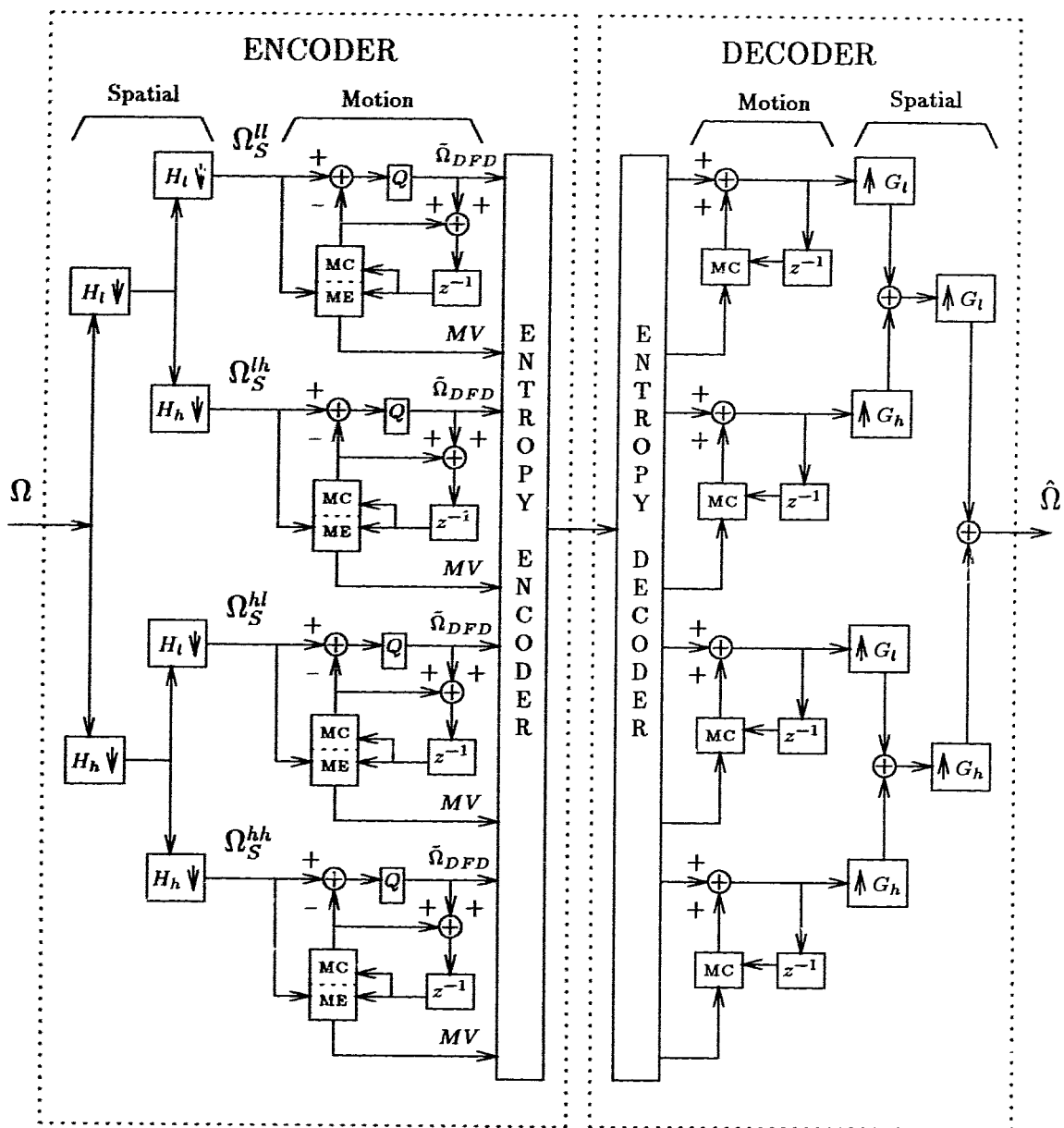


Figure 5.22: An SM Codec Block Diagram

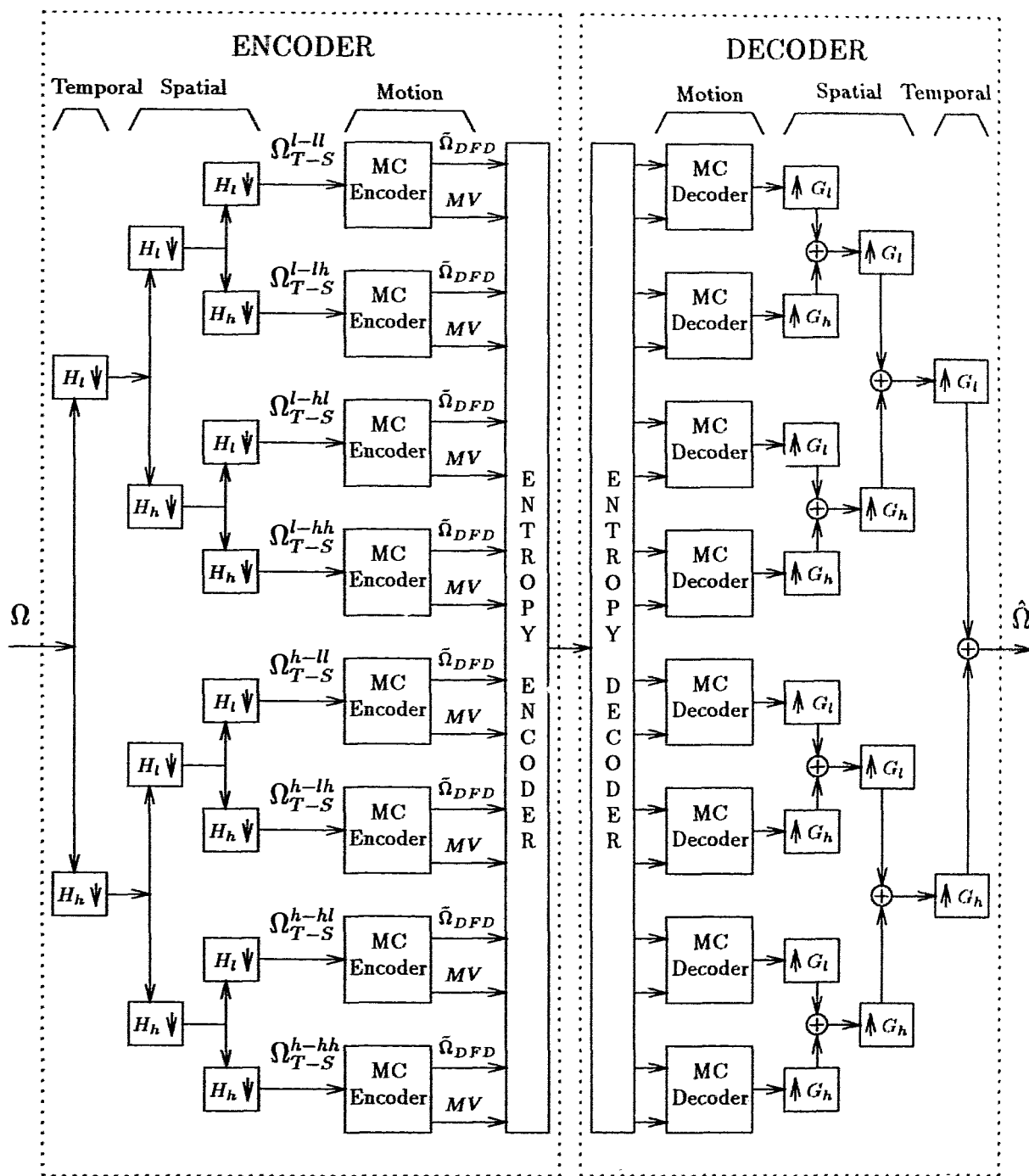


Figure 5.23: A TSM Codec Block Diagram

5.3.6 Subjective Quality Evaluations of Best Systems

Subjective quality evaluations on three codec outputs are discussed here. The three systems are the MDCT, SM, and TSM configurations. The MDCT codec results are used as a benchmark against which to rate the other two codecs. Two types of subjective evaluations were performed on the *pongi* and *missa* sequences: still frame and video. Figure 5.24-5.25 show the still frame evaluations of the regions about the man's elbow in *pongi* and the woman's face in *missa*. Following this, a discussion of subjective video evaluations is given.

In Figure 5.24, the region around the man's elbow shows motion at the twelfth reconstructed frame. In the video sequence, the man is moving his arm downward. The resolution of the postscript printed images is not high, but observations can be made. It is best to look at these images from a distance of 30 cm or so instead of up close. The subjective evaluations made here are based on evaluations of these frames on a computer screen that has much higher resolution. These frame segments are 90×70 pixels in size and an 8×8 block has dimensions of 2.7×2.7 mm. As the bit rate increases for each system, so does the quality of the images. The MDCT system introduces noticeably granular spotted-like distortions, whereas the other two subband systems introduce smoothing, low-pass filtered, distortions. At low rates, the coarsely quantized DCT coefficients in the MC feedback loop are hypothesized to cause the granular type noise, because not all the DCT basis functions are used. In the SM and TSM codecs, the DFD images are quantized only and not transformed, so the distortions in these systems are similar to those of subband codecs, i.e., low-pass filtered looking images.

In Figure 5.25, the woman's face at the twelfth reconstructed frame is shown. At low rates, the MDCT system shows blocking effects around the mouth and eyes, but the SM and TSM systems reconstruct clearer images. The SM reconstructed images are more blurry than the TSM images. For this image sequence, and at these rates, the TSM system ranks best. Next, the MDCT system is ranked better than the SM system, because the smoothing distortions about the eyes in the SM sequence at the three lowest rates are more perceptible than the MDCT's blocking distortions.

For each system and rate shown in the still frame evaluations above, subjective video evaluations were performed and are discussed here. For the *pongi* sequence, the



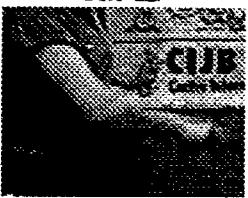

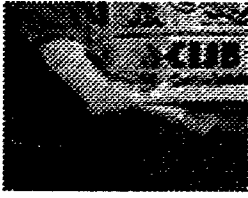

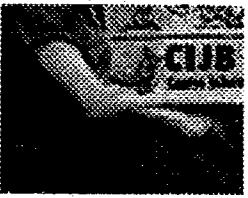
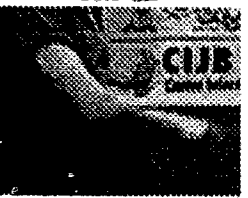

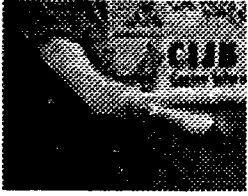
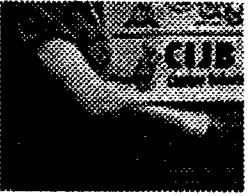
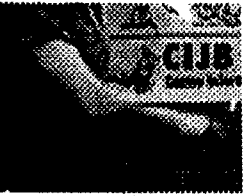
System	Average bit rate and PSNR				
MDCT	0.339 bits/pixel 26.7 dB	0.503 bits/pixel 28.7 dB	1.000 bits/pixel 34.0 dB	1.499 bits/pixel 37.8 dB	
					
	SM	0.332 bits/pixel 25.3 dB	0.489 bits/pixel 28.2 dB	0.966 bits/pixel 33.6 dB	1.470 bits/pixel 38.0 dB
					
TSM		0.343 bits/pixel 23.2 dB	0.490 bits/pixel 24.1 dB	0.986 bits/pixel 31.1 dB	1.545 bits/pixel 37.1 dB
					

Figure 5.24: Subjective Comparisons at Four Different Bit Rates for the *pongi* Sequence

MDCT system was rated best, the SM was ranked second best, and TSM the worst. In this sequence, the annoying “mosquito” background noise in the SM and TSM systems reconstructed video was the major reason they were ranked below the MDCT system. Mosquito noise is caused by quantization noise appearing and disappearing in background regions. In addition, the TSM system produced a large amount of distortion along the diagonal pong-pong table edge in front of the player. At the highest rates the system performances were ranked very close, if not equal. It may be hypothesized that if more frequency subband decompositions were to be made, the subband filtering systems would perform better at low rates, because of the increase in the number of sources, allowing for more fractional rates as in the MDCT system. For the *missa* sequence, the TSM system output was ranked best followed in descending order by the MDCT and SM systems. At the highest rate, 1.5 bits/pixel, the SM codec was ranked above the MDCT system. In this sequence, the MDCT and SM systems add significant blocking distortions in the woman’s moving upper lip, whereas



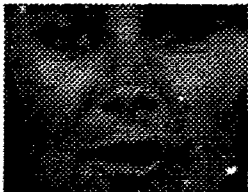









System	Average bit rate and PSNR				
MDCT	0.333 bits/pixel 39.1 dB	0.499 bits/pixel 40.1 dB	0.992 bits/pixel 42.7 dB	1.501 bits/pixel 45.5 dB	
					
	SM	0.256 bits/pixel 39.5 dB	0.509 bits/pixel 39.9 dB	0.885 bits/pixel 42.7 dB	1.417 bits/pixel 45.6 dB
					
		TSM	0.315 bits/pixel 40.1 dB	0.577 bits/pixel 41.8 dB	1.034 bits/pixel 43.9 dB
					

Figure 5.25: Subjective Comparisons at Four Different Bit Rates for the *missa* Sequence

they are imperceptible in the TSM system output. On the woman's face, the MDCT system output at low rates has distortions that look like freckles, which she does not have; in the SM system, the low-pass blurry distortions around the woman's eyes are annoying. The TSM system output is noticeably better at all four rates than the other systems.

Chapter 6

Conclusions

This thesis has presented an empirical study of digital video compression source encoders and decoders that integrate motion compensation with subband filters; it has also compared these results to those of standard coding methods, such as motion compensation – DCT coders.

The performance of twenty-two codecs was studied using three video test sequences and seven filter sets. Each video sequence contained different levels of motion. The sequences denoted as *pongi*, *sales*, and *missa*, contained high, medium, and low motion respectively. The filter sets consisted of three quadrature mirror filters (QMF), two conjugate quadrature filters (CQF), and two short kernel perfect reconstruction filters (PRF). The systems were segmented into four groups: Group 1 consisted of standard M, MDCT, DPCM, and PCM systems; Group 2, the S, T, TS, MS, and MT systems; Group 3, the SM, SM1, TM, TM1 systems; and Group 4, the TSM, TSM1, MTS, SMT, and TMS systems.

In Group 1, the rankings in descending order of performance are M – MDCT, DPCM, and PCM (note: the “–” between two systems denotes similar performances). The difference between the best and worst systems spans up to 15 dB at some rates. The computational load measured in multiplications/additions for the DPCM, M, and MDCT systems to encode a macro block of size 8×8 pixels is 192/192, 18785/36703, and 18848/36886 computations respectively.

In Group 2, the rankings, in order of performance, change from one sequence to another. For the *pongi* sequence, the system performances are ranked from best to worst as: M, MS, MDCT, MT, TS, S, and T. In *missa*, the system performances

are ranked from best to worst as: TS, MDCT, S, T, MS, M, and MT. For the *sales* sequence, the performance rankings are: MS, MDCT, TS, T, M, MT, and S. The MC-based systems performed best for the high and medium sequences and the two-dimensional subband filtering base system performed best for the low motion sequence. It was found that codecs using TS or ST configurations performed very similarly. In addition, the best filter sets for spatial and temporal subband filtering were found to be the 2-QMF and 3-5-PRF filter sets respectively. These filters were either the best, or comparable to the best, performing filter set and, because of their short kernel lengths, have low computational loads. The computational load measured in multiplications/additions for the S, T, TS, MS, and MT systems to encode an 8×8 pixel macro block is 256/192, 128/64, 640/256, 19041/36895, and 18913/36767 computations respectively.

In Group 3, the codec rankings for each video sequence follow. For the *pongi* sequence, the rankings are from best to worst: M, SM – MDCT, TM, SM1, and TM1. For the *missa* sequence, the performance rankings are: SM, TM – MDCT, SMT, TM1, and M. For the *sales* sequence, the performance rankings are: TM, TM1, SM, MDCT, SM1, and M. In *pongi* and *missa*, the results show the strengths of the SM system. The 2-QMF and 3-5-PRF temporal and spatial filter sets again performed best. The SM, SM1, TM, and TM1 system encoder computational complexity, measured in multiplications/additions per 8×8 macro block, is 5521/10479, 1572/2764, 18913/36767, and 9521/18415 respectively.

In Group 4, the codec rankings for each video sequence follow. For *pongi*, the M and MDCT codecs perform 2 to 3 dB better than any other system. Following these systems, the MTS, TMS, and SMT codecs have similar performances and are then followed by the TSM and TSM1 codecs. For *missa*, the TSM, SMT, and TSM1 codecs perform best at low rates, and the TSM performance increases by 1.5 dB over these at high rates. Following these systems, the rankings from best to worst are: TMS, MDCT, M, and MTS. For *sales*, the TMS codec performance is significantly higher than that of the other systems. Only the TSM codecs performance comes close at very low and high rates. The remaining rankings in descending order are SMT, TSM1, MDCT, and M. In the high motion *pongi* sequence, the M and MDCT codec performances dominate and, in the medium to low motion *sales* and *missa* sequences, the subband filtering systems cascaded with MC perform well. The computational load

estimated in multiplication/addition per macro block is 5649/10543 computations for the SMT and TSM encoders, 1042/1542 computations for the TSM1 encoder, and 19169/36959 computations for the TMS and MTS encoders.

In summary, it was found that for high motion video, the MC-based codecs performed best: specifically the M, SM, and MDCT systems. For the medium and low motion video sequences, the temporal and subband based codecs performed best: specifically the TSM, TM, and SM systems. The results showed that the SM codec is a good choice for high motion video; furthermore, it performs reasonably well for low motion video. Conversely, the results show that the TSM codec is a good choice for low motion video and performs reasonably well for high motion video. There is an added complexity to using the TSM codec; however, because the computational loads are dominated by MC, there are benefits to using this system. In addition, the SM and TSM codec configurations are conducive to multiresolution video systems, whereas codec configurations using MC first are not.

Subjective evaluations of the MDCT, SM, and TSM systems were performed on the reconstructed still frames and video. For the still frame evaluations, the SM codec performed best for high motion sequence frames, and the TSM codec for low motion. For the video evaluations, the MDCT system performs best for the *pongi* sequence, followed in decreasing order by the SM and TSM systems and, in the *missa* sequence, the TSM system performed best followed in decreasing order by the MDCT and SM systems.

The results presented in this work are useful to a video codec designer. The conclusions outline the strengths and weaknesses of each codec for the three video test sequences; however, more research is required to make these conclusions more general. Therefore, it is suggested that further supporting research on this topic includes a broader-based study of SM and TSM codes using more video sources, a study of SM and TSM configurations using more than the four and eight subband decompositions respectively, subjective performance measurements of multiresolution systems, and performance studies using non-separable filter sets.

Appendix A

Video Compression Chip Sets

Chip sets that implement the current video coding standards (JPEG, MPEG and H.261) require digital signal processing functions, such as input/output interfaces, color transforms, discrete cosine transforms, motion compensation codecs, quantizers, Huffman coders, run-length coders, arithmetic operators, and audio processing. Many of these coding functions require high speed processors in order to encode in real time. Included in the functionality, frame rates up to 30 frames/sec are expected and four different image or frame sizes are used. The JPEG standard uses CCIR 601 sized images (720×480 pixels); the H.261 standard uses common interchange format (CIF) images (352×288 pixels); and the MPEG standard uses either source input format (SIF) images (352×240 pixels) or NTSC images.

A list of nine integrated circuit manufacturers offering, or proposing to offer, hardware implementations of these standards is given in Table A.1.

Table A.1: Nine Chip Set Manufactures

1. Array Microsystems	VideoFLOW
2. AT&T Microelectronics	AVP-4xxx
3. C-Cube Microsystems	CL450, CL451
4. Cypress Semiconductor	—
5. Integrated Information Technology (IIT)	IIT-VP, IIT-VC
6. Intel	—
7. LSI logic	L647xx, L64112
8. SGS-Thompson	STi3240, ST54221
9. Texas Instruments	TMS320AV110, TMS6340

Appendix B

Seven Filter Impulse Responses

The impulse responses for the seven subband filter sets used in this thesis are tabulated below. The tabulation includes the analysis and synthesis low and high pass filters H_l , H_h , G_l , and G_h respectively for each filter set. The frequency response of each filter set is shown in Chapter 5, Figure 5.2.

The seven filters include three quadrature mirror filters (QMF), two conjugate quadrature filters (CQF), and two perfect reconstruction filters (PRF). The QMF filters include a 2 tap filter set defined by Smith and Barnwell (1986) and the 16b and 32c tap filter sets designed by Johnston (1980). The CQF are the 8 and 16 tap filter sets designed by Smith and Barnwell (1986). Finally, the PRF are the 3-5 and 4 tap filter sets designed by LeGall and Tabatabai (1988).

Table B.1: Smith and Barnwell (1986) 2 Tap QMF Impulse Response

tap	Analysis		Synthesis	
	low-pass	high-pass	low-pass	high-pass
1	0.5	0.5	1	-1
2	0.5	-0.5	1	1

Table B.2: LeGall and Tabatabai (1988) 3-5 Tap PRF Impulse Response

tap	Analysis		Synthesis	
	low-pass	high-pass	low-pass	high-pass
1	-0.125	0.25	0.50	0.25
2	0.250	-0.50	1.00	0.50
3	0.750	0.25	0.50	-1.50
4	0.250			0.50
5	-0.125			0.25

Table B.3: LeGall and Tabatabai (1988) 4 tap PRF Impulse Response

tap	Analysis		Synthesis	
	low-pass	high-pass	low-pass	high-pass
1	-0.25	0.125	0.25	0.50
3	0.75	-0.375	0.75	1.50
2	0.75	0.375	0.75	-0.75
4	-0.25	-0.125	0.25	-0.25

Table B.4: Smith and Barnwell (1986) 8 Tap CQF Impulse Response

tap	Analysis		Synthesis	
	low-pass	high-pass	low-pass	high-pass
1	0.034898	-0.075910	-0.151820	-0.069796
2	-0.010983	0.023900	-0.047800	-0.021966
3	-0.062865	0.357976	0.715952	0.125730
4	0.223908	-0.556857	1.113714	0.447816
5	0.556857	0.223908	0.447816	-1.113714
6	0.357976	0.062865	-0.125730	0.715952
7	-0.023900	-0.010983	-0.021966	0.047800
8	-0.075910	-0.034898	0.069796	-0.151820

Table B.5: Smith and Barnwell (1986) 16 Tap CQF Impulse Response

tap	Analysis		Synthesis	
	low-pass	high-pass	low-pass	high-pass
1	0.021936	-0.014359	-0.028718	-0.043872
2	0.001579	-0.001033	0.002066	0.003158
3	-0.060254	0.026067	0.052134	0.120508
4	-0.011891	0.006820	-0.013640	-0.023782
5	0.137538	-0.035335	-0.070670	-0.275076
6	0.057454	-0.029067	0.058134	0.114908
7	-0.321670	0.000204	0.000408	0.643340
8	-0.528720	0.295578	-0.591156	-1.057440
9	-0.295578	-0.528720	-1.057440	0.591156
10	0.000204	0.321670	-0.643340	0.000408
11	0.029067	0.057454	0.114908	-0.058134
12	-0.035335	-0.137538	0.275076	0.070670
13	-0.006820	-0.011891	-0.023782	0.013640
14	0.026067	0.060254	-0.120508	0.052134
15	0.001033	0.001579	0.003158	-0.002066
16	-0.014359	-0.021936	0.043872	-0.028718

Table B.6: Johnston (1980) 16b tap QMF Impulse Response

tap	Analysis		Synthesis	
	low-pass	high-pass	low-pass	high-pass
1	0.002898	0.002898	0.005796	-0.005796
2	-0.009972	0.009972	-0.019945	-0.019945
3	-0.001921	-0.001921	-0.003842	0.003842
4	0.035969	-0.035969	0.071937	0.071937
5	-0.016119	-0.016119	-0.032237	0.032237
6	-0.095302	0.095302	-0.190605	-0.190605
7	0.106799	0.106799	0.213597	-0.213597
8	0.477347	-0.477347	0.954694	0.954694
9	0.477347	0.477347	0.954694	-0.954694
10	0.106799	-0.106799	0.213597	0.213597
11	-0.095302	-0.095302	-0.190605	0.190605
12	-0.016119	0.016119	-0.032237	-0.032237
13	0.035969	0.035969	0.071937	-0.071937
14	-0.001921	0.001921	-0.003842	-0.003842
15	-0.009972	-0.009972	-0.019945	0.019945
16	0.002898	-0.002898	0.005796	0.005796

Table B.7: Johnston (1980) 32c Tap QMF Impulse Response

tap	Analysis		Synthesis	
	low-pass	high-pass	low-pass	high-pass
1	0.00065	-0.00065	0.00130	0.00130
2	-0.00135	-0.00135	-0.00270	0.00270
3	-0.00126	0.00126	-0.00252	-0.00252
4	0.00416	0.00416	0.00832	-0.00832
5	0.00143	-0.00143	0.00286	0.00286
6	-0.00936	-0.00936	-0.01872	0.01872
7	-0.00017	0.00017	-0.00034	-0.00034
8	0.01788	0.01788	0.03576	-0.03576
9	-0.00411	0.00411	-0.00822	-0.00822
10	-0.03116	-0.03116	-0.06232	0.06232
11	0.01447	-0.01447	0.02894	0.02894
12	0.05291	0.05291	0.10582	-0.10582
13	-0.03924	0.03924	-0.07848	-0.07848
14	-0.09980	-0.09980	-0.19960	0.19960
15	0.12847	-0.12847	0.25694	0.25694
16	0.46646	0.46646	0.93292	-0.93292
17	0.46646	-0.46646	0.93292	0.93292
18	0.12847	0.12847	0.25694	-0.25694
19	-0.09980	0.09980	-0.19960	-0.19960
20	-0.03924	-0.03924	-0.07848	0.07848
21	0.05291	-0.05291	0.10582	0.10582
22	0.01447	0.01447	0.02894	-0.02894
23	-0.03116	0.03116	-0.06232	-0.06232
24	-0.00411	-0.00411	-0.00822	0.00822
25	0.01788	-0.01788	0.03576	0.03576
26	-0.00017	-0.00017	-0.00034	0.00034
27	-0.00936	0.00936	-0.01872	-0.01872
28	0.00143	0.00143	0.00286	-0.00286
29	0.00416	-0.00416	0.00832	0.00832
30	-0.00126	-0.00126	-0.00252	0.00252
31	-0.00135	0.00135	-0.00270	-0.00270
32	0.00065	0.00065	0.00130	-0.00130

Bibliography

- Aggarwal, J. K. and N. Nandhakumar (1988, August). On the computation of motion from sequences of images - a review. *Proceedings of the IEEE* 76(8), 917-935.
- Bamberger, R. H. and M. J. T. Smith (1992, April). A filter bank for the directional decomposition of images: Theory and design. *IEEE Transactions on Signal Processing* 40(4), 882-893.
- Burt, P. and E. Adelson (1983, April). The Laplacian pyramid as a compact image code. *IEEE Transactions on Communications COM-31*, 532-540.
- Clarke, R. (1985). *Transform Coding of Images*. New York: Academic Press.
- Dubois, E. (1985, April). The sampling and reconstruction of time-varying imagery with application in video systems. *Proceedings of the IEEE* 73(4), 502-522.
- Gersho, A. and R. M. Gray (1992). *Vector Quantization and Signal Compression*. Series in Communications and Information Theory. Kluwer Academic Publishers.
- Gothé, M. and J. Vaisey (1993, May). Improving motion compensation using multiple temporal frames. In *Proceedings IEEE Pacific Rim Conference On Communications Computers and Signal Processing*, Volume 1, pp. 157-160.
- Hou, H. S. (1987, October). A fast recursive algorithm for computing the discrete cosine transform. *IEEE Transactions on Acoustics, Speech and Signal Processing* 35(10), 1455-1461.
- Huffman, D. A. (1952, September). A method for the construction of minimum redundancy codes. *Proceedings of the IRE* 40, 1098-1101.

- Jain, A. K. (1989). *Fundamentals of Digital Image Processing*. Prentice Hall.
- Jayant, N. and P. Noll (1984). *Digital Coding of Waveforms, Principles and Applications to Speech and Audio*. Prentice Hall.
- Johnston, J. (1980, April). A filter family for use in quadrature mirror filter banks. In *Proc. ICASSP'80*, pp. 291–294. IEEE.
- JPEG (1990, August 14). Joint photographic experts group: Technical specification revision 8. Technical report, ISO/CCITT.
- Karlsson, G. and M. Vetterli (1988a, July). Subband coding of video for packet networks. *Optical Engineering* 27(7), 574–586.
- Karlsson, G. and M. Vetterli (1988b). Three dimensional sub-band coding of video. In *Proceedings ICASSP'88*, pp. 1100–1103.
- Koga, T., K. Iinuma, A. Hirano, and Y. Iijima (1981). Motion compensated inter-frame coding for video conferencing. *NTC' 81 Conference Record*, G5.3.1–G5.3.5.
- LeGall, D. (1991, April). MPEG: A video compression standard for multimedia applications. *Communications of the ACM* 34(4), 47–58.
- LeGall, D. and A. Tabatabai (1988, April). Subband coding of digital images using symmetric short kernel filters and arithmetic coding techniques. In *Proc. ICASSP'88*, pp. 761–764. IEEE.
- Liou, M. (1991, April). Overview of the p×64 video coding standard. *Communications of the ACM* 34(4), 60–63.
- Netravali, A. and B. Haskell (1988). *Digital Pictures, Representation and Compression*. Applications of Communication Theory. Plenum.
- Oppenheim, A. V. and R. W. Schaffer (1989). *Discrete-Time Signal Processing*. Prentice Hall.
- Paek, H., R. C. Kim, and S. U. Lee (1992). On the motion compensated transform coding technique employing sub-band decomposition. In *Proceedings SPIE, Visual Communications and Image Processing*, Volume 1818, pp. 253–264.

- Press, W., B. Flannery, S. Teukolsky, and W. Vetterling (1988). *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press.
- Rabbani, M. and P. W. Jones (1991). *Digital Image Compression Techniques*. SPIE.
- Riskin, E. (1991, March). Optimal bit allocation via the generalized BFOS algorithm. *IEEE Transactions on Information Theory* 37(2), 400–402.
- Smith, M. J. and T. P. Barnwell (1986, June). Exact reconstruction techniques for tree structured subband coders. *IEEE Transactions on Acoustics, Speech and Signal Processing* 34(3), 434–441.
- Smith, M. J. and S. L. Eddins (1987, April). Subband coding of images with octave band tree structures. In *Proc. ICASSP'87*, pp. 1382–1385. IEEE.
- Vaisey, J., E. Yuen, and J. Cavers (1992, May). Video coding for very high rate mobile data transmission. In *Proceedings VTC'92*, pp. 259–261. IEEE.
- Vetterli, M. (1984). Multi-dimensional subband coding: Some theory and algorithms. *Signal Processing* 6, 97–112.
- Walker, D. and K. Rao (1984, October). Improved pel-recursive motion compensation. *IEEE Transactions on Communications* 32(10), 1128–1134.
- Wang, L. and M. Goldberg (1989, December). Progressive image transmission using vector quantization on images in pyramid form. *IEEE Transactions on Communications* 37(12), 1339–1349.
- Woods, J. and S. O'Neil (1986, October). Subband coding of images. *IEEE Transactions on Acoustics, Speech and Signal Processing* 34, 1278–1288.
- Woods, J. W. and T. Naveen (1992, July). A filter based bit allocation scheme for subband compression of HDTV. *IEEE Transactions on Image Processing* 1(3), 436–440.
- Zaccarin, A. and B. Liu (1992, March). Fast algorithms for block motion estimation. In *Proceedings ICASSP'92*, pp. III-449 – III-452.