PYRAMID CODING FOR IMAGE AND VIDEO COMPRESSION

by

David Ian Houlding

B.Sc. (Electronic Engineering) University of Natal,
Durban, Natal, South Africa,
1992

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

SIMON FRASER UNIVERSITY
June, 1994

APPROVAL

Name: David Ian Houlding

Degree: Master of Applied Science

Title of thesis : Pyramid Coding for Image and Video Compression

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

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

Dr. Vladimir Cuperman
Professor of Engineering Science, SFU
Supervisor

Dr. Brian Funt
Professor of Computing Science, SFU
Examiner

Date Approved: 27 / June / 1994

PARTIAL COPYRIGHT LICENSE

Title of Thesis/Project/Extended Essay

"Pyramid Coding for Image and Video Compression"

Author:

(signature)

David Houlding
(name)

June 17, 1994
(date)

Abstract

This research investigates the use of pyramid coding in various digital image and video compression applications, the progressive transmission of images, and the efficient recovery of motion information from video sequences. Through the use of feedback, pyramid coding allows both flexibility in the choice of filters, and the use of quantization noise feedback. While the filters have a significant effect on the properties of the generated pyramid, quantization feedback allows more control over the distortion introduced indirectly into the reconstructed image through separate quantization of the pyramid subimages.

After a frequency domain analysis of pyramid coding, a new class of decimation and interpolation filters is developed. The use of both different filters, and quantization feedback in various applications is then explored. This is followed by proposals regarding possible choices of generation schemes and filters for the efficient implementation of pyramid coding in these applications. Finally, the performance of the pyramid codec is evaluated against both optimal methods, where appropriate, and other proposed suboptimal methods.

For example, a filter pair is proposed for lossless image coding that not only leads to the generation of low entropy pyramids, but also allows a given image to be represented as a subsampled pyramid of the same number of pixels. Lossless compression ratios significantly higher than in the case of related techniques are achieved by this form of pyramid coding, while generally at a fraction of the computational cost. Appropriate filters are also proposed for the generation of pyramids suitable for the recovery of motion information from video sequences. Using hierarchical motion field recovery, based on pyramids generated with these filters, it is shown that a motion compensation scheme achieves a performance close to that achieved when based on the optimal exhaustive search technique, while at a small fraction of the computational cost.

Acknowledgements

Foremost, I would like to thank Dr. Jacques Vaisey for his guidance and supervision received during the course of my graduate research at Simon Fraser University. I am also grateful to the Canadian Cable Labs, NSERC's IOR (Industrially Oriented Research) program, and the Simon Fraser University Graduate Fellowship Committee for their generous financial support.

Contents

List of Tables

List of Figures

List of Abbreviations

ADPCM	adaptive DPCM
ATM	asynchronous transfer mode
BFOS	Breiman, Friedman, Olshen and Stone
bpp	bits per pixel
BPS	bits per second
CCDC	Channel Compatible DigiCipher (HDTV standard)
CCITT	International Telegraph and Telephone Consultative Committee
codec	coder / decoder combination
DCT	discrete cosine transform
DFD	displaced frame difference
DOG	difference of Gaussians
DPCM	differential pulse code modulation
FIR	finite impulse response
Flops	floating point operations
HDTV	high definition television
HVS	human visual system
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
JBIG	Joint Bi-level Image Experts Group
JPEG	Joint Photographic Experts Group
KLT	Karhunen-Loève transform
MAD	mean absolute difference
MC	motion compensation
MEP	minimal entropy pyramid
MF	motion field

MPEG	Moving Picture Experts Group
MSE	mean squared error
NMSE	normalized mean squared error
NTSC	National Television System Commission
PB	passband
pdf	probability distribution function
PP	paired pyramid
PSNR	peak signal to noise ratio
QMF	quadrature mirror filter
RDP	reduced difference pyramid
SB	stopband
SE	squared error
SNR	signal to noise ratio
SSE	sum of squared error

Chapter 1

Introduction

Information, the communication or reception of knowledge or intelligence, is the essential commodity that enables us to learn and make decisions. However, to communicate and learn from information, we must have some method of transmitting and storing it. This requires the use of various communications channels, for example radio, television or telephone and storage systems, for example electronic memory, magnetic media or optical media. To make the best use of communication channels and storage systems, it is necessary to maximize their efficiency by minimizing the transmission and storage of redundant, or useless, information. Increased efficiency translates into lower bandwidth requirements in a communication channel, or lower memory requirements in a storage system. Source coding encompasses a variety of techniques which strive to minimize information redundancy present in a source and thereby improve communication or storage efficiency in various applications.

There exist two broad classifications of source coding, namely lossless and lossy source coding. Lossless source coding does not introduce any distortion into a source, and finds applications in, for example, the compression of medical images. Due to the constraint of perfect reconstruction of the original signal, lossless coding typically achieves compression ratios up to three. On the other hand, lossy source coding is capable of achieving much higher compression ratios, but may introduce some distortion into the reconstructed signal. Due to its generally higher compression capability, lossy coding finds application in, for example, image databases, teleconferencing and video-on-demand services.

Pyramid coding is a source coding technique that finds application in both lossless and lossy image and video compression. This thesis presents research directed at tailoring and integrating pyramid coding into new schemes for image and video compression.

1.1 Image and Video Sources

Natural images are typically comprised of smoothly varying, continuous tone "surfaces", and consequently consist of mostly low spatial frequency information. This classification encompasses the majority of "real world" scenes. Source coding algorithms for 8 bpp greyscale natural images, for example see Figure B.1, have been investigated in this research. However, these algorithms may easily be extended for the coding of higher rate or color natural images.

A video signal is composed of a sequence of frames, spaced at regular time intervals, as shown in Figure 1.1. Temporal lowpass filtering of the *human visual system* (HVS) leads to the perception of apparent motion from small changes in successive frames in the video sequence. There exist two broad classes of video, namely progressive and interlaced video. In progressive video, each frame consists of a complete image, while in interlaced video, each frame consists of an even or odd field. The alternating even or odd fields in interlaced video in turn contain either the even or odd numbered rows in the image frames respectively. For example, the NTSC television standard, widely used in North America, defines an 394×525 interlaced video signal with a rate of 60 frames per second. On the other hand, the CCDC HDTV standard defines a 720×1280 progressive video signal, also with a rate of 60 frames per second. For 24 bpp color, such a progressive signal represents an uncompressed rate of 1.3×10^9 BPS, and highlights the need for efficient source coding. In this research, source coding algorithms for progressive video signals consisting of 8 bpp greyscale frames, for example see Figure B.2, have been investigated, although they may easily be extended to higher rate or color progressive video.

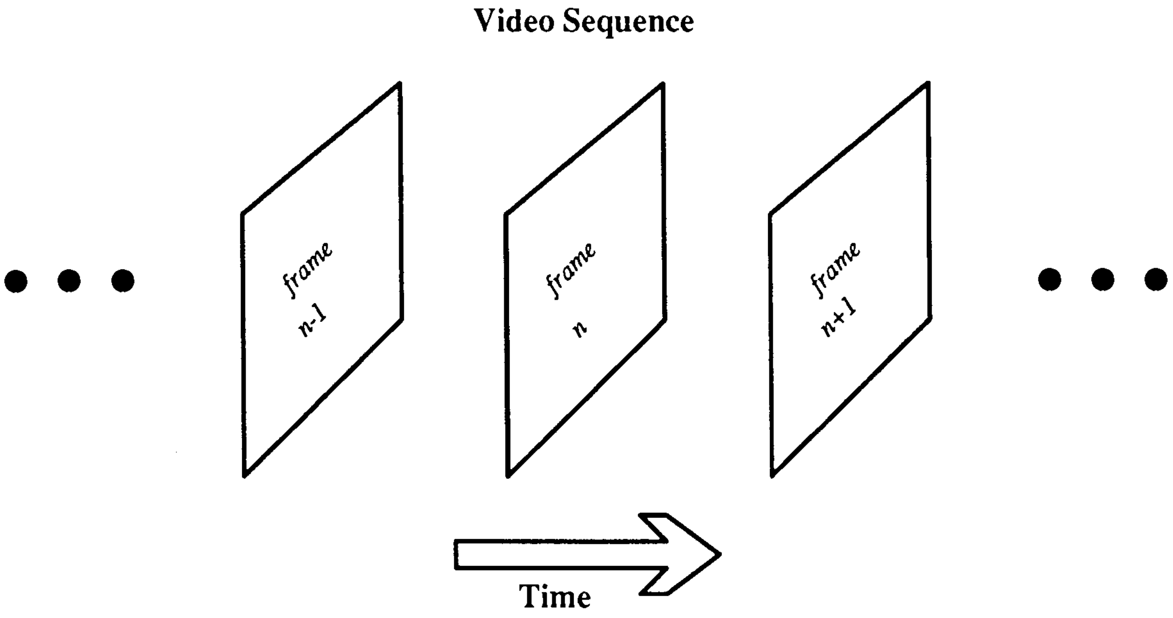


Figure 1.1: Video Sequence Diagram

1.2 Pyramid Coding

Pyramid coding is a source coding technique that was first proposed by Burt and Adelson (1983) for the compression [1] of still images. Subsequently, research done by Ho and Gersho (1989b), Torbey and Meadows (1989) and Wang and Goldberg (1989a, 1989b, 1991) has lead to better ways of applying pyramid coding in the compression of still images. Furthermore, research done by Stiller and Lappe (1991) and Uz, Vetterli, and LeGall (1991) has lead to new applications of pyramid coding in video compression. Through pyramid coding, an image may be represented in an alternative hierarchical form as an image pyramid. An example of a three level image pyramid is shown in Figure 1.2. An image pyramid consists of a set of subimages of various sizes, the largest of which is the same size as the original image from which the pyramid was constructed. Each subimage represents a level of the pyramid, and contains a different portion of the spatial frequency spectrum of the original image. In order of decreasing size, the pyramid subimages contain mostly highpass, bandpass and lowpass information of the original image respectively.

[1] Compression refers to the process by which data is represented in a more concise, compact form.

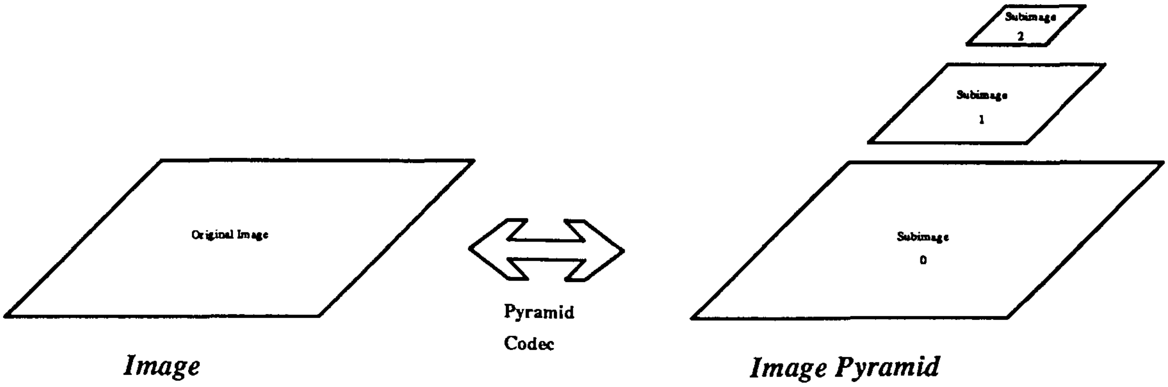


Figure 1.2: Pyramid Codec Dataflow Diagram

1.2.1 Advantages

Pyramid coding, like subband coding (discussed later in Section 3.1), has a number of advantages over other related source coding techniques.

Generally, most energy in natural images is concentrated at low spatial frequencies (Section 1.1). Consequently, the spatial frequency decomposition done by pyramid coding on an image leads to a set of subimages in which the variance increases with decreasing size. This property may be exploited through separate coding of the subimages, where the larger, low variance subimages are usually coarsely quantized to achieve a significant compression.

Filters used in the pyramid generation process have a significant effect on the properties of the generated pyramid (Burt and Adelson 1983). Unlike subband coding, feedback in the pyramid generation process allows flexibility in the choice of these filters so that pyramid coding can be tailored to generate pyramids suitable for a particular application.

In quantization of the pyramid, each of the pyramid subimages is quantized separately. Without feedback, this leads to an accumulation of quantization noise at low spatial frequencies (Uz, Vetterli, and LeGall 1991). This is undesirable, since the HVS is most sensitive to noise in this region (Kronander 1989). However, in the case of pyramid coding, quantization noise introduced to the pyramid subimages may be fed back through the pyramid generation process (Wang and Goldberg 1989a), allowing more control of the distortion introduced indirectly into the image reconstructed from the pyramid. This is not possible in subband coding where no such feedback path exists.

Since the *human visual system* (HVS) is most sensitive to low spatial frequency information, the significance of information contained in the pyramid subimages generally decreases with increasing size. For instance, it is possible to transmit the pyramid subimages over a channel in order of increasing size, and decreasing importance, so that the most significant information is sent first to be subsequently refined with the arrival of less important information. This technique, which is particularly useful in the transmission of images over low bandwidth channels, is called *progressive transmission* (Burt and Adelson 1983). Also, since the pyramid subimages contain different portions of the spectrum of the original image, the dependence of the sensitivity of the HVS on spatial frequency may be conveniently exploited through separate coding of the pyramid subimages.

Since the subimages have different resolutions, pyramid coding may be classified as a multiresolution technique. The hierarchy of image information contained in the image pyramid permits the use of various computationally efficient "coarse-to-fine" image analysis strategies.

1.2.2 Disadvantages

However, there exist two main disadvantages to pyramid coding. Firstly, the sum of the pixels in the pyramid subimages generally exceeds the number of pixels in the original image, except in the case of a specialized pyramid coding scheme developed later in this thesis. In subband coding however, the sum of the pixels in the subband images is the same as the number of pixels in the original image. Secondly, the pyramid generation and image reconstruction processes generally have a relatively high computational cost, when compared to other similar source coding techniques.

These disadvantages make the task of applying pyramid coding in image and video compression more challenging, and are amongst some of the issues addressed in this research.

1.3 Motion Compensation

Video sequences typically contain a considerable amount of temporal redundancy as a result of the fact that neighboring frames [2] generally do not differ significantly. Various methods have been proposed to exploit this temporal redundancy in order to achieve a significant compression of the video information. The most successful and widely applied of these is *motion compensation* (MC). MC represents a given frame as an interpolation of neighboring frames, plus an interpolation error. The details of this temporal interpolation are described using a *motion field* (MF) that specifies the way in which the given frame differs from the neighboring frames. The error in the interpolation when compared to the original frame, which is commonly referred to as the *displaced frame difference* (DFD), may then be used together with the MF to reconstruct the video sequence exactly. If the MF is sufficiently accurate in describing the way in which the given frame differs from the neighboring frames, the DFD will have low energy [3] and may be efficiently coded, leading to a good compression. However, accurate calculation of the MF often requires enormous computational effort, and in practice suboptimal techniques must therefore be used. For example, the hierarchical structure of pyramid coding allows computationally efficient "coarse-to-fine" strategies to be implemented in the calculation of the MF.

1.4 Overview

This thesis documents research done to explore the advantages and overcome the disadvantages of pyramid coding with the aim of making it more useful in practical image and video compression applications.

After the development of new classes of decimation and interpolation filters, based on a frequency domain analysis of pyramid coding, the significance of the filters in pyramid codecs for various applications is analyzed. This is followed by proposals regarding possible choices of generation schemes and filters for the efficient implementation of pyramid coding in these applications. For example, an efficient new lossless image coding technique, based on pyramid coding, is developed. Various bit

[2] A frame refers to a single image in a video sequence.

[3] The energy of an image is represented by its variance.

allocation schemes are investigated for use in lossy pyramid codecs that employ quantization feedback. Progressive transmission of images using pyramids is also explored. Finally, techniques are proposed to exploit the hierarchical structure of the image pyramid in video applications, allowing more accurate MF estimates to be calculated at lower computational cost.

Chapter 2 briefly reviews basic aspects of source coding after which Chapter 3 discusses the details of pyramid coding. Applications of pyramid coding in image and video compression are then investigated in Chapter 4 and Chapter 5 respectively. Finally, Chapter 6 draws some conclusions regarding the use of pyramid coding in practical applications and suggests some potential areas for future research.

Chapter 2

Foundations In Source Coding

2.1 Source and Channel Coding

Generally, two types of signal coding are implemented in a practical codec, namely source and channel coding. Source coding attempts to represent data in an alternative form where significant information is concentrated in some sense and can be transmitted or stored more efficiently. Channel coding on the other hand, encompasses various techniques used to encode data prior to transmission or storage in order to make it more robust to noise and other degradations. In the general model of a codec shown in Figure 2.1, source and channel coding may be approximated as independent, cascaded processes. In such a model, separate optimization of the source and channel codecs leads to an optimal combined codec. This model has proven to be a reasonable approximation in practice and has been used in this research, where separate optimization of the source codec in image and video applications is investigated. However, in some cases, joint source and channel coding may provide an overall performance gain. Optimization of the channel codec is outside the scope of this research.

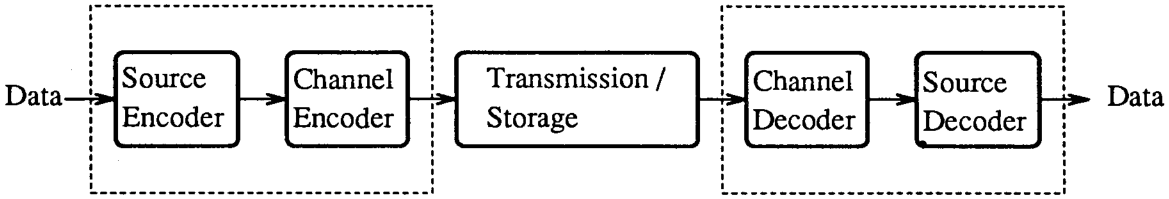


Figure 2.1: General Codec Model

2.2 Information Redundancy

The objective of source coding is to represent data in an alternative, more compact form, or "compress" it. This is done by identifying and eliminating information redundancy, which is generally present in data in one or both of two forms. Firstly, statistical redundancy results from inter-symbol[1] correlation in the data. A signal that does not possess any statistical redundancy has an autocorrelation function that is an impulse. Secondly, perceptual redundancy is present in data as a result of information that is not significant in the application for which it is intended. For example, in images and video, perceptual redundancy exists as a result of information content that is not visible or significant to the HVS.

2.3 The Source Codec

The source coder and decoder can be broken down as shown in Figure 2.2 (Sezan and Lagendijk 1993). Initially, the source coder represents the input signal in an alternative form, referred to as the *representation coefficients*, in which significant information is concentrated in some way. In the next stage, bits are allocated to the coefficients according to their importance in determining the quality of the reconstructed signal. After bit allocation, the coefficients are quantized, to produce the *quantization indices.* Finally, codewords, the components of which are generally bits, are uniquely assigned to the quantization indices to produce the *bitstream.* The decoder implements the inverse transformations of the above three stages in the reverse order to produce the *reconstructed signal.* For simplicity, Figure 2.2 neglects overhead information needed to implement the representation, quantization and codeword assignment stages and their inverses. Generally, this information represents only a small fraction of the total bitstream though.

2.3.1 Representation

The representation stage of the source coder can be divided into the four broad classes of predictive, transform, spectral decomposition and model based coding methods.

[1] A data symbol refers to the basic unit of a data signal. For example, in images, a symbol refers to an individual pixel.

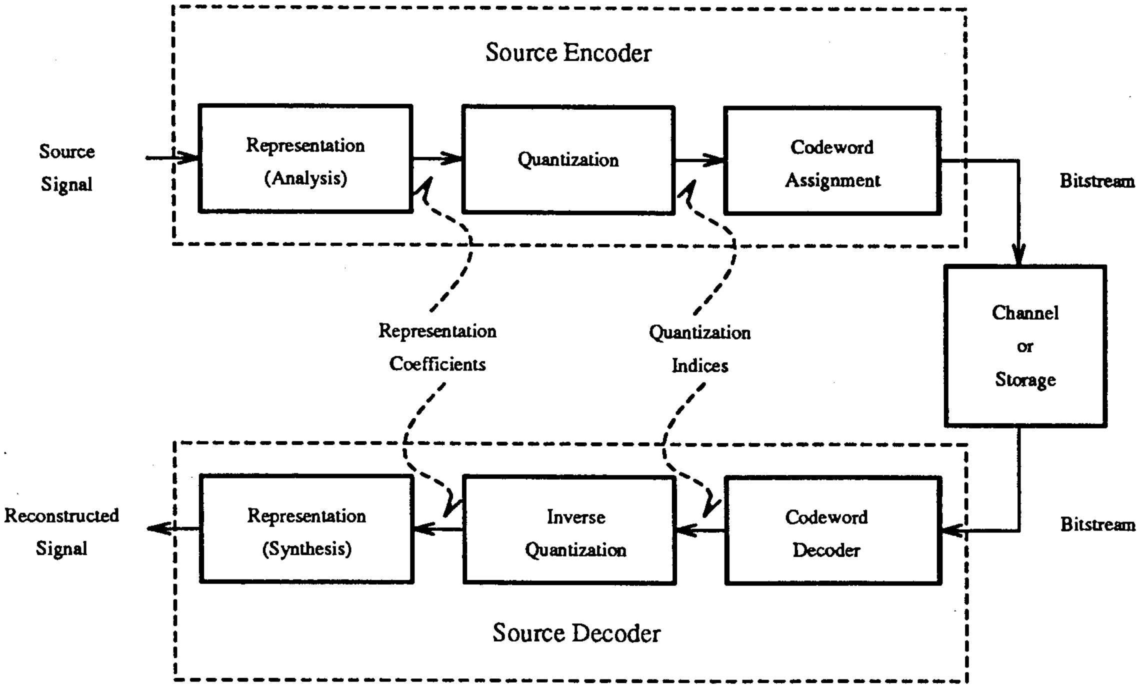


Figure 2.2: Source Encoder and Decoder Functional Breakdown

For any technique, a tradeoff is made in computational complexity versus efficiency of the representation, in terms of describing the most useful information of the source in the smallest number of coefficients.

Predictive Coding

In image and video coding, often there exists a significant amount of correlation between neighboring pixels in the spatial, color or temporal dimensions. This enables predictive coding to estimate the value of a given pixel from the values of other pixels that are close by in some sense (Gersho and Gray 1992). A source may therefore be represented by a prediction equation and prediction error. Since the prediction equation is most commonly linear with only a few terms, and the decorrelated prediction error has low energy, this results in a significant concentration of source information. The most common implementation of predictive coding is *differential pulse code modulation* (DPCM). In DPCM, the coefficients of the linear prediction equation are calculated in advance, based on measured or estimated statistics of the source. It therefore assumes stationarity of the source. Alternatively, this technique has also been implemented adaptively in *adaptive DPCM* (ADPCM), enabling the predictor to "follow" slow changes in the statistics of the source. Since predictive coding techniques rely on perfect synchronization of the encoder and decoder, they are sensitive to channel errors, and are usually "reset" for resynchronization during operation. Often, calculation of the coefficients of the predictor equation introduces considerable computational complexity into this class techniques and has limited their use in some applications. Note that prediction in the temporal domain, often applied in video sequence source coding, is called MC. In later discussions of MC, some important differences between DPCM and MC prediction will become clear.

Transform Coding

Transform coding encompasses a variety of linear techniques used to represent a source in an alternative form in which significant data is concentrated in only a few transform coefficients, thus enabling compression (Netravali and Haskell 1988). The optimal transformation, in terms of decorrelating the transform coefficients, is the *Karhunen-Loève Transform* (KLT). However, the KLT is source dependent and therefore computationally expensive. A more practical and source independent transform, which has been widely implemented in practice, is the *discrete cosine transformation* (DCT) (Clarke 1985). However, the DCT, and other suboptimal transform coding techniques, are typically block based and therefore exhibit annoying blocking artifacts at low rates.

Spectral Decomposition Coding

Coding methods implementing spectral decomposition of the source include subband (Woods and O'Neil 1986), (Woods 1991) and pyramid coding (Burt and Adelson 1983), (Akansu 1992). These multiresolution techniques initially perform an octave subband decomposition of the spectrum of the original image into a number of subbands or subimages, each of which can then be coded according to its specific characteristics. Where the spectral energy of a source is non-uniformly distributed, as is generally the case in images and video (see Section 1.1), spectral decomposition coding leads to a set of subbands/subimages with generally different variances. Signal energy is therefore concentrated in high variance subbands/subimages, and low variance subbands/subimages may often be coarsely quantized. The HVS is most sensitive to low spatial frequency content in an image or video source. Most perceptually significant data is therefore generally found in the low spatial frequency subbands/subimages. It should be noted, however, that the application of MC prior to spectral decomposition coding in video compression confuses this relationship between spectral content of the subbands and the sensitivity of the HVS (Gothe 1993). These techniques have the advantage that, in the decoding stage, filtering helps to spread quantization noise over many pixels so that undesirable effects such as blocking and contouring are less apparent in the reconstructed images.

Model Based Coding

Model based coding methods, for example fractal coding (Barnsley and Hurd 1993), attempt to represent source in a much higher level symbolic form with a small number of coefficients, potentially allowing a very high data compression to be achieved. However, these techniques typically require a relatively high computational cost, particularly in the encoder, limiting their application to date.

2.3.2 Quantization

Often, the representation coefficients in Figure 2.2 consist of a large range of floating point values. In order to store or transmit the coefficients, it is necessary to assign unique indices to each coefficient. However, this assignment generally results in a very high bitstream rate, and an inefficient source coder. It is therefore necessary to reduce the number of unique coefficients by approximating ranges of coefficients by single coefficients. This process is called quantization. Before quantization, bit allocation must be done to determine the structure of the quantizers. A common algorithm used to determine the optimal bit allocation for a set of quantizers, resulting in the lowest overall distortion for a selection of rates, is the *Breiman, Friedman, Olshen and Stone* (BFOS) algorithm (Riskin 1991). Alternatively, the simpler, suboptimal Greedy bit allocation algorithm (Gersho and Gray 1992) may be used for the same purpose.

Quantization (Gersho and Gray 1992) is a nonlinear transformation whereby a block of input symbols, each of which is an element of some (possibly infinite) input set, is assigned a quantization index. Inverse quantization then consists of uniquely assigning a block of output symbols, each of which is generally also an element of the input set, to each quantization index. In scalar quantization, each symbol is quantized independently and the block dimension is therefore one. Vector quantization involves quantization with larger block dimensions. While vector quantization has the ability to exploit intersymbol correlation in order to achieve a better performance, scalar quantizer design and implementation is computationally cheaper and therefore more widely implemented in practice. Another advantage of vector quantization is that it allows the assignment of fractional bit rates to the coefficients. Whereas, in scalar quantization, each unique coefficient must be assigned a unique quantization index, and codeword of integer length greater than one.

A scalar quantizer may be decomposed into a set of quantization cells, the i^{th} cell of which is shown in Figure 2.3. In the quantization operation, all input values in the range $[x_i, x_{i+1})$ are assigned a unique quantization index, for example i. During inverse quantization, this index is assigned a predetermined output value y_i, generally within the range $[x_i, x_{i+1})$. In Figure 2.3, x_i, y_i, x_{i+1} and Δ are called the lower boundary, output, upper boundary and step size of the quantization cell respectively. Quantizer cells may be classified as granular or overload as shown in Figure 2.4. Granular cells are bounded, but overload cells lack one boundary, which is effectively set to infinity.

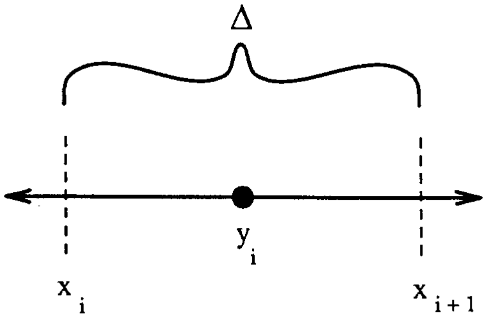


Figure 2.3: Quantization Cell

Scalar quantizers may be classified as either uniform or nonlinear. In the case of nonlinear quantizers, Δ may vary from cell to cell, along with the relative position of y_i in the cell. However, in the case of uniform quantizers, Δ is constant for all granular cells and and y_i is always the midpoint of the cell. The performance of uniform quantization, when combined with appropriate codeword assignment, is asymptotically equal to that of optimal nonlinear quantization with increasing bit rate (Jayant and Noll 1984). Uniform quantizers are simple to design and implement, and the complete quantizer structure may be specified with only a few coefficients. Uniform quantizers are also often designed to be symmetrical about the mean of the signal to be quantized, based on the assumption of a symmetric signal pdf. This is a valid assumption, for example, in the case of the pyramid subimages that generally have pixel pdfs resembling Laplacian distributions. A useful parameter called the load fraction of the quantizer may be defined. The load fraction β for a uniform quantizer with the structure in Figure 2.4, symmetrical about $x_{N/2}$ which is defined to be the signal mean, is calculated as

$$\beta = \frac{\sigma}{y_{max}} \tag{2.1}$$

where σ is the standard deviation of the signal to be quantized, and y_{max} is the absolute distance from $x_{N/2}$ to y_{N-1} or y_0. Since any distortion in the reconstructed signal generally results mainly from this stage of the source codec, careful bit allocation and quantizer design is necessary for good overall source codec performance.

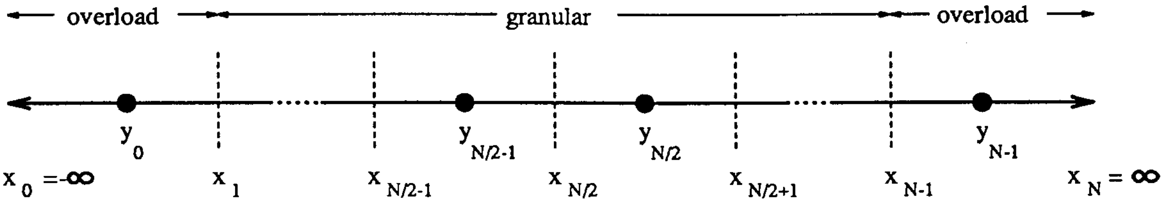


Figure 2.4: Quantizer

2.3.3 Codeword Assignment

In this stage, the quantization indices are uniquely assigned codewords, the components of which are usually represented, for transmission purposes, as binary numbers. In fixed rate codeword assignment, all codewords have the same length, while in variable rate codeword assignment, codewords may have different lengths. Fixed rate codeword assignment allows for both a simpler implementation, and easier synchronization of the encoder and decoder, an important consideration in error prone channels. On the other hand, variable rate codeword assignment requires the use of more sophisticated synchronization techniques and buffers. The optimal (lowest) average rate for any assignment in which a codeword is assigned to each quantizer index, is defined by the zeroth order entropy H of the quantizer indices (A.8). Fixed rate codeword assignment is only optimal for this stage if each quantization index is equiprobable. However, for nonuniform quantizer index pdfs, appropriate variable rate codeword assignment results in an overall average rate closer to the optimal entropy lower bound. For this reason, variable rate codeword assignment is sometimes referred to as entropy coding. Since this stage of the source coder introduces no distortion into the reconstructed signal, it is also often referred to as noiseless coding (Gersho and Gray 1992).

Codewords may be assigned to groups of quantization indices, for example in *run-length coding* (Gersho and Gray 1992) or *arithmetic coding* (Langdon and Rissanen 1981). Alternatively, they may be assigned to each quantization index independently, for example in *Huffman coding* (Huffman 1952). In the former case, blocks of quantizer indices must be processed at a time, while Huffman coding allows instant encoding and decoding of the quantizer indices, one at a time. However, in Huffman coding, each quantizer index must be assigned a unique codeword of integer length greater than zero. This constraint may cause the overall rate to be significantly higher than the ideal entropy lower bound in applications where the quantizer index pdf is highly nonuniform. Codewords may also be assigned using adaptive techniques, capable of following variations in the statistics of the quantizer indices. Two examples such techniques are *adaptive Huffman coding* (Gersho and Gray 1992) and *Ziv-Lempel coding* (Ziv and Lempel 1978).

2.4 Lossless / Lossy Source Coding

There exist two broad classifications of source coding techniques, namely *lossless* and *lossy coding.*

In lossless coding, the reconstructed signal matches the original signal exactly. Due to this constraint, lossless coding schemes only exploit statistical redundancy in the original signal in order to achieve compression, and omit the quantization stages of the codec. Lossless coding commonly achieves compression ratios C (A.14) in the approximate range $C \in [1,3]$ (Gersho and Gray 1992). Where a source contains statistical redundancy in, for example, temporal or color dimensions, as well as the spatial dimension, higher lossless compression ratios can generally be achieved. Lossless coding finds application in, for example, medical and satellite imaging, where no loss of quality is tolerable for legal or other reasons.

For higher compression ratios, lossy coding may be used. However, some distortion may be added to the reconstructed signal by the source coder, the severity of which generally increases with increasing compression ratio. This enables lossy coding to exploit statistical as well as perceptual redundancy in the original signal. Similar increases in lossy compression ratios are possible for sources containing additional correlations in, for example, temporal or color dimensions. Lossy coding, due to its typically higher compression ratios, is more widely applied in, for example, image and video compression for the mass media market, including television, video conferencing and image databasing.

2.5 Source Codec Performance

The performance of a source codec is most commonly measured as its rate-distortion performance. In image coding, the rate is defined in terms of *bits per pixel* (bpp) (A.4) while distortion refers to the "quality" of the reconstructed signal when compared to the input signal. Ideally, in image and video source coding applications, quality would reflect the subjective visual quality of the reconstructed data as perceived by the end viewer. Since the viewer in a vast majority of such applications is human, and the HVS is far from well understood, this subjective quality is not easily quantified. It is therefore common practice to substitute some objective, mathematically tractable measure of quality for the ideal, for example the *mean squared error* (MSE), *normalized mean squared error* (NMSE), *signal to noise ratio* (SNR) or *peak signal to noise ratio* (PSNR) (A.10,A.11,A.12,A.13). In the interest of comparing results presented in this research with other related research, these objective measures have been used.

It should be noted that, given some measure of distortion, the rate-distortion performance of a codec is dependent not only on the methods employed by the codec, but also on the data. In order to optimize the performance of a codec for a certain application, it is therefore necessary to match the codec to the specific characteristics of the data. This requires choosing a coding technique that is capable of exploiting particular statistical and perceptual information redundancies present in the data. Alternatively, the codec can be made adaptive so that it can "follow" changes in the characteristics of the data during operation.

There may also be a need for the rate of the source codec to be variable in some applications, for example in communications over a mobile channel with time variant characteristics. In such an application, the source codec must drop or increase its rate during operation to limit the error probability. In this case, it is desirable that as the rate of the codec becomes more constrained, the quality of the data output from the decoder degrades gracefully. Graceful degradation requires that the distortion introduced into the data by the codec does not contain annoying artifacts.

A further consideration in, for example, "packetized" ATM networks is the ability to prioritize packets of data (Chen, Sayood, and Nelson 1992), since this allows the network to alleviate congestion by discarding low priority packets. For an efficient communication system, such prioritization of a given packet must be closely linked to the perceptual importance of the source coding coefficients contained within. A source coding technique that facilitates such prioritization is therefore desirable in these applications.

Chapter 3

Principles of Pyramid Coding

As discussed in Section 2.3.1, subband and pyramid coding are examples of spectral decomposition coding. After a brief outline of subband coding, in order to contrast some of the unique features of pyramid coding, both lossy and lossless schemes for various pyramid coding applications will be examined in detail.

3.1 Subband Coding

In efficient source coding, it is desirable to concentrate useful information from a source (see Section 2). Natural images typically contain most of their energy at low spatial frequencies (see Section 1.1). Therefore, it is possible to concentrate the image information to some extent by decomposing the spatial frequency spectrum of the image source into into subbands (see Section 2.3.1). The subband containing the low spatial frequency detail usually contains the most important image information, to the HVS, in a concentrated form. This is the principle behind subband coding (Woods and O'Neil 1986), a more detailed description of which follows.

The basic subband codec analysis-synthesis iteration is shown in Figure 3.1. In the coder, the original signal s(n) is filtered by a pair of analysis filters $\mathbf{A}_{\mathrm{l}}(\omega)$ and $\mathbf{A}_{\mathrm{h}}(\omega)$ with lowpass and highpass responses respectively. Subsequently, the two filter outputs are subsampled by a factor of two ($\downarrow \mathbf{2}$), quantized ($\mathbf{Q}$) and entropy coded ($\mathbf{E}$) for transmission or storage. The decoding iteration proceeds in the reverse order, beginning with entropy decoding ($\mathbf{E}^{-1}$), followed by inverse quantization ($\mathbf{Q}^{-1}$), upsampling ($\uparrow \mathbf{2}$) and finally, synthesis filtering with the lowpass and highpass filters

$S_l(\omega)$ and $S_h(\omega)$ respectively. The summed outputs of the synthesis filtering operations form the reconstructed signal $\hat{s}(n)$. The iteration in Figure 3.1 decomposes $s(n)$ into two subbands, each subsampled by a factor of two, with the same total number of samples as in the original signal. For decompositions into a larger number of subbands, the iteration is repeated with $s(n)$ as the subsampled output of $A_l(\omega)$. In summary, subband coding performs an octave subband decomposition (see Figure 3.8) on the spectrum of the original signal.

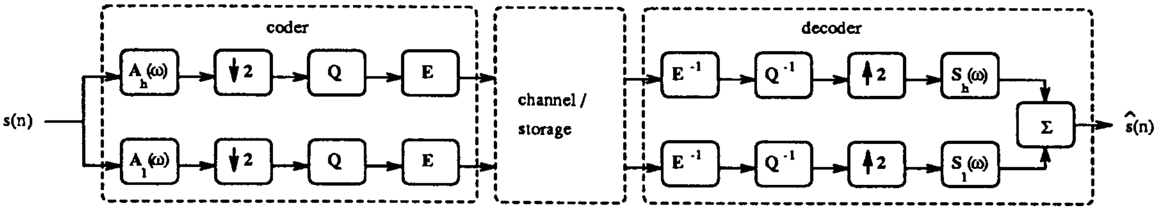


Figure 3.1: The Basic Subband Codec Analysis-Synthesis Iteration

Ideally, the analysis filters $A_l(\omega)$ and $A_h(\omega)$ are "brickwall" lowpass and highpass filters, with cutoffs at $\frac{\pi}{2}$, unity gain in the passband, and infinite attenuation in the stopband. These filters would allow the outputs to be subsampled by a factor of two without any aliasing. Similar ideal synthesis filters $S_l(\omega)$ and $S_h(\omega)$ would allow the unwanted "interpolation image", resulting from the upsampling operation, to be stopped perfectly so that the original signal can be reconstructed exactly. In practice however, these filters cannot be realized, and noise from both aliasing and "interpolation images" gets added in the codec operation. However, by imposing a set of constraints on the analysis-synthesis filter pairs, it is possible to ensure aliasing free, perfect reconstruction of the original signal so that, in the absence of quantization, $\hat{s}(n) = s(n)$. Quadrature mirror filters (QMF) (Esteban and Galand 1977) are an example of a filter class that satisfy these constraints. The constraints for QMF filters are summarized below.

$$\begin{aligned} |A_l(\omega)| &= |A_h(\omega+\pi)| \\ S_l(\omega) &= 2A_l(\omega) \\ S_h(\omega) &= -2A_h(\omega) \\ |A_l^2(\omega)| + |A_h^2(\omega)| &= 1 \end{aligned} \tag{3.1}$$

In practice, these constraints can be closely approximated using appropriate FIR filters that have a modest size. In image and video coding applications, the subband decomposition may be done in two, or sometimes three dimensions (Gothe 1993) by applying the one dimensional QMF filters (3.1) separably. It is also possible to implement subband decomposition using nonseparable filters, although the design of such filters is a complex problem.

In real applications, filtering becomes a problem near the signal limits, for example the edges of an image, where samples are not present. This problem can be handled in at least three different ways. Firstly, the unknown samples can assume the value of some constant. Secondly, the signal extremes can be joined, for example the image can be joined at its top and bottom, and left and right in a way that is analogous to circular convolution (Oppenheim and Schafer 1989). Thirdly, Smith and Eddins (1987) have shown that when the unknown samples are assigned values by reflection about the signal extremes, the energy of the subbands is generally reduced. See further in Section 3.5.2 for a more detailed discussion of filtering edge effects.

Subband coding, unlike pyramid coding, has the advantage that the total number of samples in the subbands equals the number of samples in the original signal. However, the analysis-synthesis filter pairs are rigidly constrained and may not be altered to suit a particular application, without losing the ability to reconstruct the original signal perfectly. In addition, it is not possible to implement feedback in the subband coding iteration. Consequently, separate quantization of the subbands may lead to an undesirable accumulation of quantization noise in the reconstructed signal.

On the other hand, it is possible to use feedback in the pyramid coding iteration, allowing both flexibility in the choice of filters and more control of the distortion introduced indirectly into the reconstructed signal, as a result of separate quantization of the subimages. These, and other advantages of pyramid coding have been explored in this research, and will be discussed in more detail in the following chapters.

3.2 Background

The first scheme used to generate image pyramids is shown in Figure 3.2. After lowpass filtering an image, subsampling is done, generally by a factor of two horizontally and vertically, to produce a "coarse" quarter size version of the original image called the decimated image. The original and decimated image then form the base and top subimages of a two level pyramid. For pyramids with more than two levels, the generation iteration can be re-applied with the previous decimated image as the original image and so forth. The purpose of the lowpass filtering operation is to attenuate the high spatial frequencies, above $\pi/2$, in the original image to reduce the effects of aliasing during subsampling. Burt and Adelson (1983) proposed that, since the impulse response of their lowpass filter (for use in the generation iteration in Figure 3.2) resembles a Gaussian function, the generated pyramid should be referred to as the *Gaussian pyramid.* Since then, *Gaussian pyramid* has been be adapted to refer to all pyramids generated with the iteration in Figure 3.2, a convention that has been used in this research.

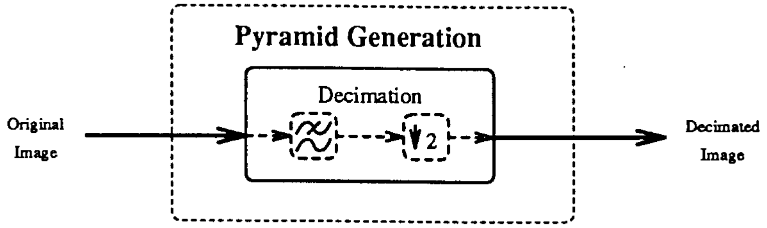


Figure 3.2: The Gaussian Pyramid Generation Iteration

Although the Gaussian pyramid has found extensive application in various, efficient coarse-to-fine image analysis tasks, particularly in the realm of computer vision, it has not been applied directly for source coding of images or video. This is due to the fact that the Gaussian pyramid is a redundant image representation, in that the base of the pyramid is the same as the original image, and the other pyramid subimages represent overhead, or redundant, information. There also exists a large amount of correlation both within the subimages, and between the subimages of the Gaussian pyramid. However, as will be shown in Chapter 5, indirectly, the Gaussian pyramid finds important application in the source coding of video.

Later, it will be shown that, through the use of different generation schemes, it is possible to generate other types of image pyramids with different properties.

3.3 The Pyramid Codec

The implementation details of both the pyramid coder (also referred to as the pyramid generation scheme) and the pyramid decoder (also referred to as the image reconstruction scheme) differ in the case of lossy and lossless coding.

3.3.1 Lossy Pyramid Coding

The basic pyramid generation and image reconstruction iterations for lossy pyramid coding can be represented as shown in Figure 3.3 (Burt and Adelson 1983). In the **decimation** process, an image is first lowpass filtered by a two dimensional decimation filter, after which it is subsampled by a factor of two horizontally and vertically. The decimated image contains the low spatial frequency information present in the original image, but has only 1/4 the number of pixels. The purpose of the decimation filter is to reduce aliasing by attenuating spatial frequencies above $\pi/2$ in the original image prior to subsampling. The **interpolation** process consists of upsampling the decimated image by a factor of two horizontally and vertically. Unknown pixels in the upsampling operation are padded with zeros. This is followed by lowpass filtering of the upsampled image using a two dimensional interpolation filter. The purpose of the interpolation filter is to amplify the spatial frequencies below $\pi/2$ in the upsampled image that are attenuated as a result of the upsampling operation, and to attenuate the spatial frequencies above $\pi/2$ representing the unwanted "interpolation image".

The **pyramid generation** iteration consists of decimation, followed by interpolation and subtraction of the interpolated image from the original. The difference and decimated subimages form the base and top of a two level pyramid, and contain mostly high and low spatial frequency information respectively. Larger pyramids can be generated by cascading the generation iteration so that the top of the pyramid formed from the first iteration is the original image for the second iteration and so forth. Where the decimation and interpolation filter impulse responses resemble Gaussian functions, the generation iteration in Figure 3.3 approximates the application of a *difference of Gaussian* (DOG) operator (Burt and Adelson 1983). In turn, the DOG operator has been shown to approximate the Laplacian operator (Marr 1982), used widely in image processing for edge detection. Therefore, Burt and Adelson (1983) proposed that this type of pyramid be called the *Laplacian pyramid*, a convention that has been followed in this research.

Generally, the pixels of the pyramid subimages consist of a wide range of floating point values, as a result of the filtering operations, and must be quantized (**Q**) before entropy coding (**E**) can be efficiently applied. In the **image reconstruction** iteration, entropy decoding ($\mathbf{E}^{-1}$) followed by inverse quantization ($\mathbf{Q}^{-1}$) are first used to recover the pyramid base and top subimages, after which the reconstructed image, an approximation of the original image, can be calculated by interpolating the pyramid top and adding it to the pyramid base. The reconstruction iteration may be cascaded in a similar manner to the generation iteration for larger pyramids.

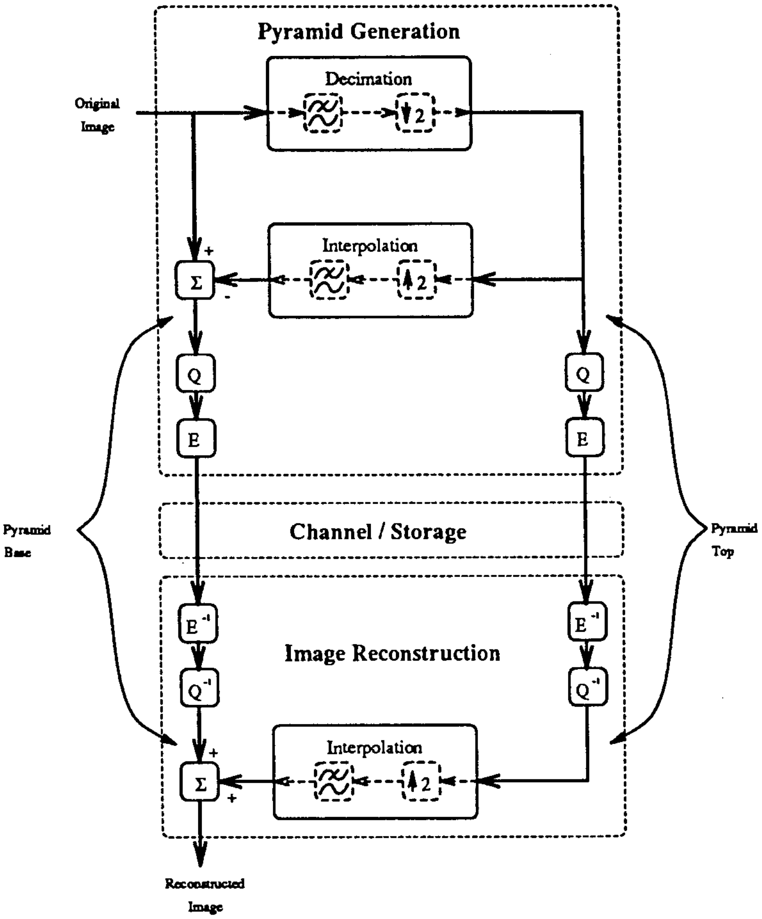


Figure 3.3: General Lossy Pyramid Coding and Decoding Iterations

3.3.2 Lossless Pyramid Coding

Quantization of the pyramid subimages in the codec shown in Figure 3.3 leads to undesirable distortion in the reconstructed image. However, the pyramid subimages cannot usually be entropy coded efficiently without first being quantized, as discussed in Section 3.3.1. To solve this dilemma, feedback can be used in the pyramid generation scheme, enabling quantization of the subimages, while preserving the ability to reconstruct the original image exactly. Two such schemes for applying quantization feedback are now discussed.

Lossless Pyramid Coding with Quantization Feedback

The first scheme for lossless pyramid coding, employing quantization feedback, is shown in Figure 3.4. This scheme was first proposed by Wang and Goldberg (1989a). Since quantization of the pyramid top is done prior to interpolation and calculation of the pyramid base, quantization noise introduced into the pyramid top is fed back to the pyramid base. This allows the range of pixel values in the pyramid top to be restricted through quantization, leading to efficient entropy coding of the pyramid top, while preserving the ability to reconstruct the original image exactly. However, such a generation scheme generally results in a pyramid base with a wide range of floating point pixel values, making it unsuitable for efficient entropy coding. As well, the pyramid base cannot be quantized without introducing distortion into the original image, since there is no way to implement quantization feedback from the pyramid base. This subimage is also the largest of the pyramid, and therefore contributes significantly to the total rate. Consequently, such a scheme is not practical for efficient lossless pyramid coding. It does however have important advantages when used as a lossy codec, as will be discussed further in Section 3.6.3.

Practical Lossless Pyramid Coding with Quantization Feedback

Figure 3.5 illustrates a second scheme for lossless pyramid generation, which is more practical than the technique proposed by Wang and Goldberg (1989a) (see Figure 3.4) since it also allows efficient entropy coding of the pyramid base. This scheme is a generalization of that proposed by Goldberg and Wang (1991), where the post-decimation and post-interpolation quantizers are constrained to quantize the floating

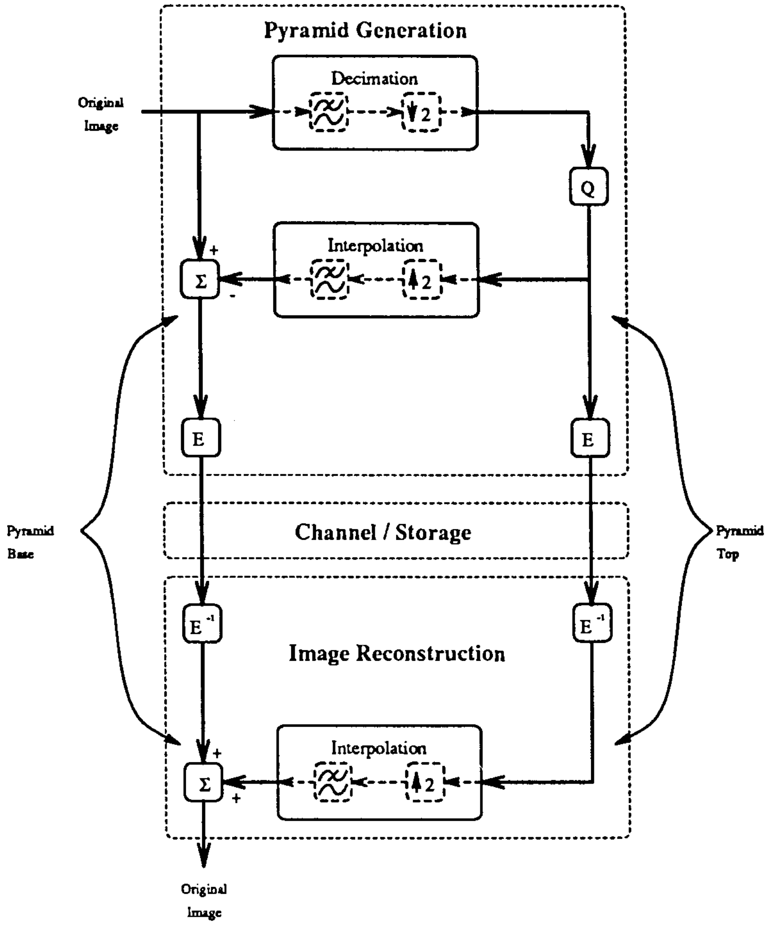


Figure 3.4: Lossless Pyramid Coding and Decoding Iterations Using Quantization Feedback

point values to the nearest integer.

In the scheme shown in Figure 3.5, the range of floating point pixel values in the pyramid subimages is restricted by the use of quantization after both the decimation and interpolation filtering operations. For example, if the original image consists of integer pixel values, and the quantizers in the decimation and interpolation processes quantize each floating point pixel value to the nearest integer, then the pyramid subimages will consist of only integer pixel values. This allows efficient entropy coding of all the pyramid subimages, while preserving the ability to reconstruct the original image exactly.

Note that the quantization stages in Figure 3.5 have been incorporated into the decimation and interpolation processes since these operations may be carried out implicitly by constraining the filtering operations to output only a finite set of pixel values. It will shown later in Section 4.1.1 that, through appropriate choice of pyramid filters, it is possible to implement finite field arithmetic in the subtraction and addition operations in the pyramid codec shown in Figure 3.5. This allows the pixels in the pyramid subimages to be represented in the same number of bits as the pixels in the original image.

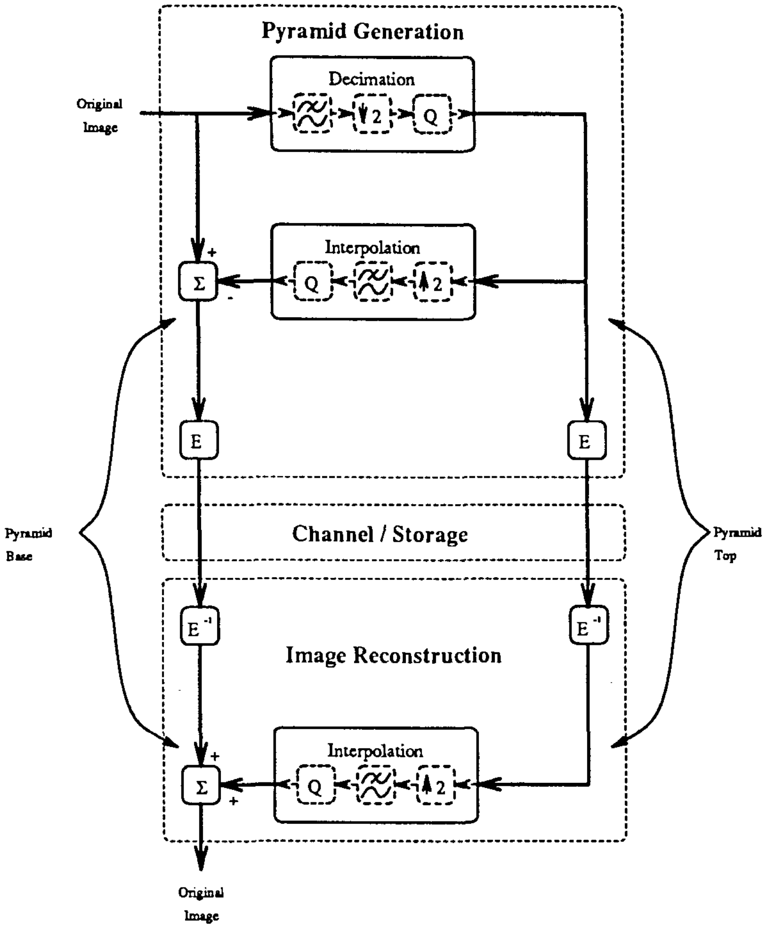


Figure 3.5: General Lossless Pyramid Coding and Decoding Iterations

3.4 Pyramid Coding in the Frequency Domain

The effect of subsampling and upsampling by a factor of two on the spectrum of the original signal is illustrated in Figure 3.6 (Oppenheim and Schafer 1989). Clearly, the ideal decimation filter, needed to avoid aliasing in the subsampling operation, is

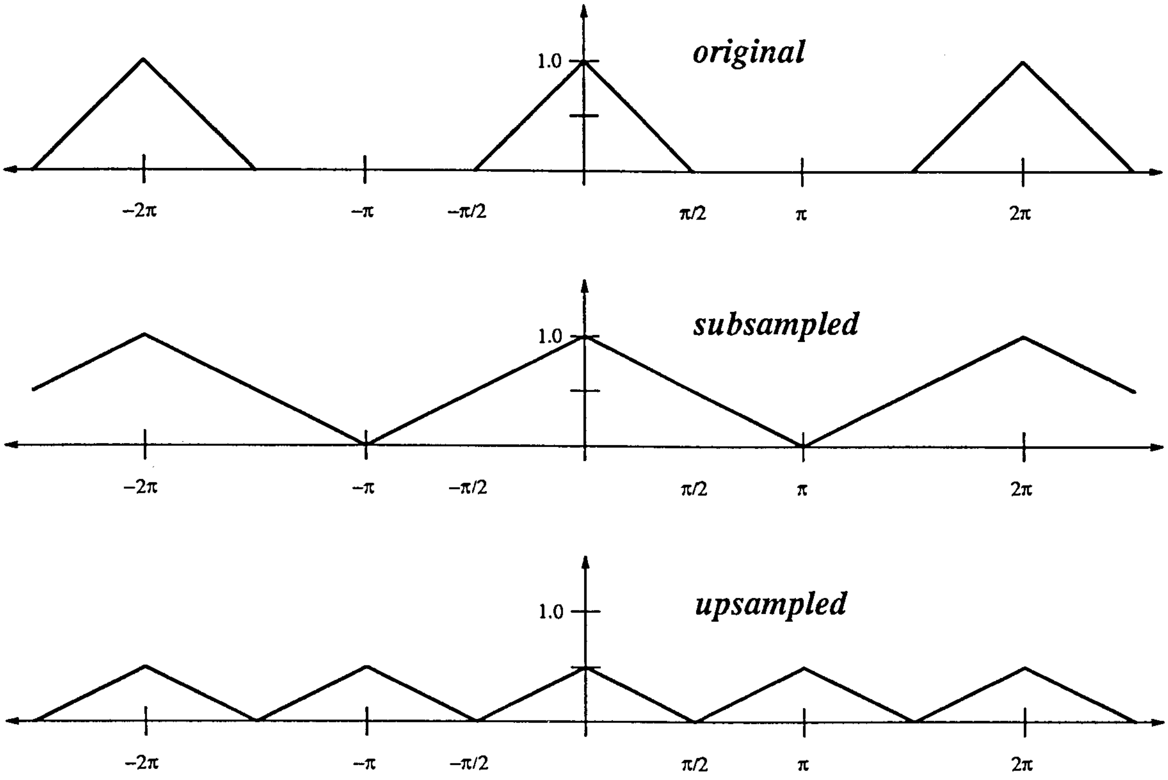


Figure 3.6: Subsampling and Upsampling in the Frequency Domain

a "brickwall" lowpass filter with a cutoff at $\pi/2$, unity gain in the *passband* (PB), and infinite attenuation in the *stopband* (SB). The ideal interpolation filter, needed to stop the unwanted "interpolation image" resulting from the upsampling operation, is identical, except it must have a gain of two in the passband. These ideal filters are shown in Figure 3.7. Pyramid generation with these ideal filters would result in a perfect octave subband decomposition of the spectrum of the original image into the pyramid subimages, in that there would be no "overlap" in the frequency ranges assigned to the different pyramid subimages. For example, a five level pyramid generated with these ideal filters would result the frequency decomposition shown in Figure 3.8, where region **0** would be assigned to the pyramid base through to region **4** which would be assigned to the pyramid top.

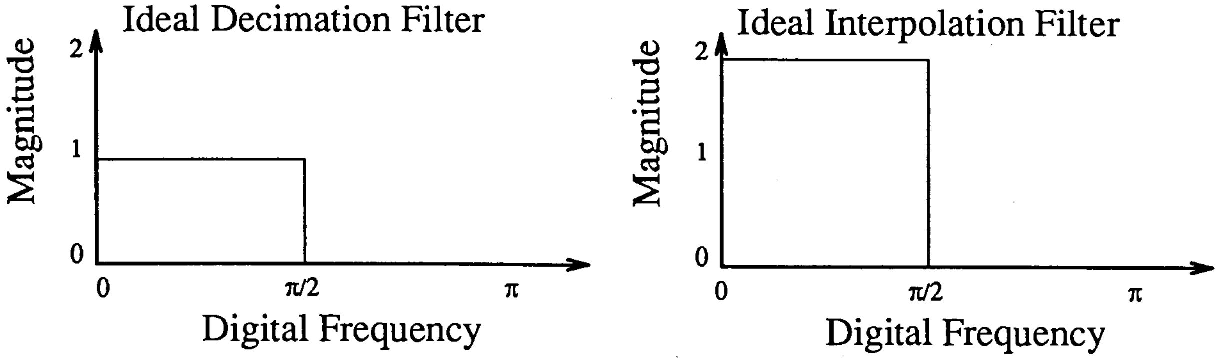


Figure 3.7: Ideal Pyramid Filter Frequency Responses

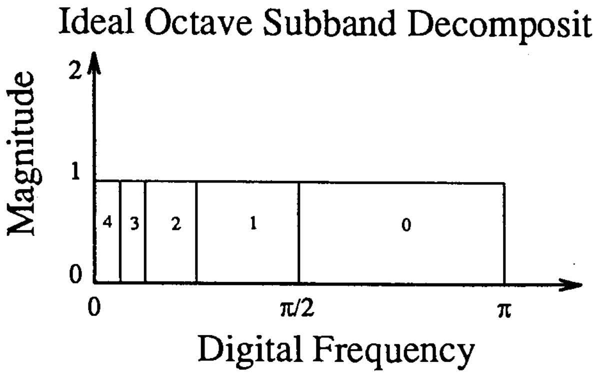


Figure 3.8: Ideal Octave Subband Decomposition for a Five Level Pyramid

3.5 Pyramid Filters

Feedback in the pyramid generation process permits the use of a large variety of decimation and interpolation filters. This can be contrasted with subband coding, where the choice of such filters is severely constrained (see Section 3.1). The choice of these pyramid filters has a significant influence on the properties of the generated pyramid (Burt and Adelson 1983), and allows pyramid coding to be "tailored" to suit a particular application.

In spite of this flexibility, and the significance of the filters on the properties of the pyramid, the majority of previous pyramid coding research has not properly addressed the issue of what filters to choose for the generation of pyramids suitable for a given practical application. There are two known exceptions to this. Firstly, Burt and Adelson (1983) defined a new filter class in which a particular filter was found to lead to the generation of the lowest entropy pyramid, for that class. However, the same filter class was used for both the decimation and interpolation filtering operations (see Figure 3.3.1). As was shown in Section 3.4, the objectives of the decimation and interpolation filtering operations are different, and this is an unnecessary constraint. Secondly, Gurski, Orchard, and Hull (1992) presented a technique for calculation of the optimal pyramid filters that minimize the variance of the pyramid base in each generation iteration. However, these optimal filters are image dependent, and the proposed technique therefore involves a vast amount of computation to calculate the optimal filters prior to generation of the pyramid.

This research includes the derivation of a more practical, image independent class of decimation and interpolation filters for use in a variety of different pyramid coding applications. Although it is not possible to realize both the ideal passband and stopband responses of a pyramid filter[1] together, as defined in Section 3.4, they can be realized separately. The design of a practical pyramid filter therefore requires a tradeoff to be made between the ideal passband and stopband responses. Using this frequency domain approach to pyramid coding, a new class of pyramid filters has been designed.

3.5.1 A New Class of Pyramid Filters

Generally, the more complexity allowed in a filter, the closer it can be made to approximate an ideal response. However, for any practical filter implementation, there is a limit in computational cost, and indirectly the complexity. It is desirable for the pyramid filters to also have the property of zero phase, so that they have a real frequency response and do not impart a "shift" to any frequency in the image during the filtering operations. A zero phase filter may be conveniently realized as a symmetric, noncausal *finite impulse response* (FIR) filter (Elliot 1987). These requirements, together with considerations of stability and simplicity in design, lead to the choice of 5×5 coefficient FIR pyramid filters in this new class. The pyramid filters have also been constrained to be separable, allowing the two dimensional filtering operation to be realized more simply as two separate one dimensional filtering operations. Separability is a property that allows a given pyramid filter to be completely specified using only one coefficient, as will be demonstrated further in the derivation of this

[1] Pyramid filter is a collective term referring to either the decimation or interpolation filter used in the pyramid codec.

new filter class.

A separable pyramid filter $\mathbf{F}$ can be represented as the outer product of a 1×5 FIR filter $\mathbf{f}$

$$\begin{aligned}\mathbf{F} &= \mathbf{f}^{\mathbf{T}} \times \mathbf{f} \\ &= [f_{-2}\ f_{-1}\ f_0\ f_1\ f_2]^T \times [f_{-2}\ f_{-1}\ f_0\ f_1\ f_2]. \end{aligned} \tag{3.2}$$

The resulting filter $\mathbf{F}$ has a rectangular region of support in the two dimensional frequency domain, determined by the one dimensional frequency response of the filter $\mathbf{f}$ from which it is calculated. Therefore, design of a two dimensional pyramid filter may be simplified to the design of a one dimensional filter. Consequently, the coefficients of $\mathbf{f}$ may be represented as

$$\begin{aligned}\mathbf{f} &= [\ f_{-2}\ \ f_{-1}\ \ f_0\ \ f_1\ \ f_2\ ] \\ &= [\ c\ \ b\ \ a\ \ b\ \ c\ ], \end{aligned} \tag{3.3}$$

where a, b, and c are real constants, so that the one dimensional frequency response can be found by

$$\begin{aligned}\mathcal{F}[\mathbf{f}] &= \sum_{k=-2}^{2} f_k\, e^{-jwk} \\ &= a + 2b\cos(w) + 2c\cos(2w). \end{aligned} \tag{3.4}$$

The decimation and interpolation filters, designed in the rest of this section, have been constrained to have perfect DC[2] responses, leading to near ideal low spatial frequency responses, since this is where most energy is found in natural images. The performance of a pyramid filter in a given frequency band is calculated as the *squared error* (SE) of its response, when compared to the ideal response (see Section 3.4) in that band. For design of both the decimation and interpolation filters, the one dimensional passband and stopband filter performances are first represented as a function of the filter coefficients a, b and c in (3.3). The overall squared error filter performance may then be represented as a combination of the passband and stopband performances, where the filter coefficients are constrained to yield a filter with a perfect DC response. Finally, minimization of this overall performance with respect to the filter coefficients leads to a system of equations that can be used to completely specify a 5×5 pyramid filter, given a single parameter that specifies the tradeoff in passband versus stopband performance for that filter.

[2] DC refers to the zero frequency part of the spatial frequency spectrum.

The Decimation Filter

The *passband performance* (PB_{SE}) of the decimation filter is given by

$$\begin{aligned}
PB_{SE} &= \int_0^{\pi/2} (\mathcal{F}[\mathrm{f}] - 1)^2 dw \\
&= \int_0^{\pi/2} (a + 2b\cos(w) + 2c\cos(2w) - 1)^2 dw \\
&= \int_0^{\pi/2} 2c^2\cos(4w)dw \\
&\quad + \int_0^{\pi/2} 4bc\cos(3w)dw \\
&\quad + \int_0^{\pi/2} (4ac + 2b^2 - 4c)\cos(2w)dw \\
&\quad + \int_0^{\pi/2} (4ab + 4bc - 4b)\cos(w)dw \\
&\quad + \int_0^{\pi/2} (a^2 + 2b^2 + 2c^2 - 2a + 1)dw \\
&= -\frac{4bc}{3} + 4ab + 4bc - 4b + \frac{\pi}{2}(a^2 + 2b^2 + 2c^2 - 2a + 1), \qquad (3.5)
\end{aligned}$$

and the *stopband performance* (SB_{SE}) by

$$\begin{aligned}
SB_{SE} &= \int_{\pi/2}^{\pi} (\mathcal{F}[\mathrm{f}] - 0)^2 dw \\
&= \int_{\pi/2}^{\pi} (a + 2b\cos(w) + 2c\cos(2w))^2 dw \\
&= \int_{\pi/2}^{\pi} 2c^2\cos(4w)dw \\
&\quad + \int_{\pi/2}^{\pi} 4bc\cos(3w)dw \\
&\quad + \int_{\pi/2}^{\pi} (4ac + 2b^2)\cos(2w)dw \\
&\quad + \int_{\pi/2}^{\pi} (4ab + 4bc)\cos(w)dw \\
&\quad + \int_{\pi/2}^{\pi} (a^2 + 2b^2 + 2c^2)dw \\
&= \frac{4bc}{3} - 4ab - 4bc + \frac{\pi}{2}(a^2 + 2b^2 + 2c^2). \qquad (3.6)
\end{aligned}$$

To satisfy the requirement that the decimation filter have an ideal DC response, the following constraint is introduced

$$a + 2b + 2c = 1. \qquad (3.7)$$

This leads to the overall performance of the decimation filter, J_{SE}, which may be represented as a linear combination of the passband and stopband performances in (3.5) and (3.6), constrained by (3.7)

$$\begin{aligned}
J_{SE} &= \sigma_d PB_{SE} + (1-\sigma_d)SB_{SE} + \lambda(a+2b+2c-1) \\
&= \frac{4bc}{3} - 4ab - 4bc + \frac{\pi}{2}(a^2+2b^2+2c^2) \\
&\quad +\sigma_d(-\frac{8bc}{3} + 8ab + 8bc - 4b - \pi a + \pi/2) \\
&\quad +\lambda(a+2b+2c-1). \qquad (3.8)
\end{aligned}$$

$$0 \;\le\; \sigma_d \;\le\; 1$$

In this equation, σ_d is a parameter used to specify the relative importance of the passband versus the stopband performances in the filter design, while λ is a Lagrange multiplier used to incorporate constraint (3.7) into the optimization. Differentiating J_{SE} with respect to a, b, c and λ gives

$$\begin{aligned}
\frac{dJ_{SE}}{da} &= -4b + \pi a + 8\sigma_d b - \pi\sigma_d + \lambda \\
\frac{dJ_{SE}}{db} &= \frac{4c}{3} - 4a - 4c + 2\pi b - \frac{8\sigma_d c}{3} + 8\sigma_d a + 8\sigma_d c \\
\frac{dJ_{SE}}{dc} &= \frac{4b}{3} - 4b + 2\pi c - \frac{8\sigma_d b}{3} + 8\sigma_d b + 2\lambda \\
\frac{dJ_{SE}}{d\lambda} &= a + 2b + 2c - 1. \qquad (3.9)
\end{aligned}$$

Setting these derivatives equal to zero then leads to the following system of equations

$$\begin{bmatrix} \pi & (8\sigma_d-4) & 0 & 1 \\ (8\sigma_d-4) & 2\pi & (16\sigma_d-8)/3 & 2 \\ 0 & (16\sigma_d-8)/3 & 2\pi & 2 \\ 1 & 2 & 2 & 0 \end{bmatrix}^{-1} \begin{bmatrix} \pi\sigma_d \\ 4\sigma_d \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} a \\ b \\ c \\ \lambda \end{bmatrix}. \qquad (3.10)$$

Given a choice of σ_d, this system of equations can be solved to completely specify a decimation filter.

Interpolation Filter

The *passband performance* (PB_{SE}) of the interpolation filter is given by

$$PB_{SE} \;=\; \int_0^{\pi/2} (\mathcal{F}[\mathrm{f}] - 2)^2 dw$$

$$\begin{aligned}
&= \int_0^{\pi/2}(a+2b\cos(w)+2c\cos(2w)-2)^2dw \\
&= \int_0^{\pi/2} 2c^2\cos(4w)dw \\
&\quad + \int_0^{\pi/2} 4bc\cos(3w)dw \\
&\quad + \int_0^{\pi/2} (4ac+2b^2-8c)\cos(2w)dw \\
&\quad + \int_0^{\pi/2} (4ab+4bc-8b)\cos(w)dw \\
&\quad + \int_0^{\pi/2} (a^2+2b^2+2c^2-4a+4)dw \\
&= -\frac{4bc}{3}+4ab+4bc-8b+\frac{\pi}{2}(a^2+2b^2+2c^2-4a+4), \qquad (3.11)
\end{aligned}$$

and the *stopband performance* (SB_{SE}) by

$$\begin{aligned}
SB_{SE} &= \int_{\pi/2}^{\pi}(\mathcal{F}[\mathrm{f}]-0)^2dw \\
&= \frac{4bc}{3}-4ab-4bc+\frac{\pi}{2}(a^2+2b^2+2c^2), \qquad (3.12)
\end{aligned}$$

as in (3.6). To satisfy the requirement that the interpolation filter have an ideal DC response, the following constraints are introduced

$$a+2c=1, \qquad 2b=1, \qquad (3.13)$$

This leads to the overall performance of the interpolation filter J_{SE}, which may be represented as a linear combination of the passband and stopband performances in (3.11) and (3.12) respectively, constrained by (3.13)

$$\begin{aligned}
J_{SE} &= \sigma_i PB_{SE}+(1-\sigma_i)SB_{SE}+\lambda_1(a+2c-1)+\lambda_2(2b-1) \\
&= \frac{4bc}{3}-4ab-4bc+\frac{\pi}{2}(a^2+2b^2+2c^2) \\
&\quad +\sigma_i(-\frac{8bc}{3}+8ab+8bc-8b-2\pi a+2\pi) \\
&\quad +\lambda_1(a+2c-1)+\lambda_2(2b-1). \qquad (3.14)
\end{aligned}$$

$$0 \leq \sigma_i \leq 1$$

Again, σ_i is the parameter used to specify the tradeoff to be made between the passband and stopband performances, while λ_1 and λ_2 are Lagrange multipliers used to incorporate the constraints in (3.13) into the optimization. Differentiating J_{SE} with respect to a, b, c, λ_1 and λ_2 gives

$$\begin{aligned}
\frac{dJ_{SE}}{da} &= -4b + \pi a + 8\sigma_i b - 2\pi\sigma_i + \lambda_1 \\
\frac{dJ_{SE}}{db} &= \frac{4c}{3} - 4a - 4c + 2\pi b - \frac{8\sigma_i c}{3} + 8\sigma_i a + 8\sigma_i c - 8\sigma_i + 2\lambda_2 \\
\frac{dJ_{SE}}{dc} &= \frac{4b}{3} - 4b + 2\pi c - \frac{8\sigma_i b}{3} + 8\sigma_i b + 2\lambda_1 \\
\frac{dJ_{SE}}{d\lambda_1} &= a + 2c - 1 \\
\frac{dJ_{SE}}{d\lambda_2} &= 2b - 1. \qquad (3.15)
\end{aligned}$$

Setting these derivatives equal to zero then leads to the following system of equations

$$\begin{bmatrix} \pi & (8\sigma_i - 4) & 0 & 1 & 0 \\ (8\sigma_i - 4) & 2\pi & (16\sigma_i - 8)/3 & 0 & 2 \\ 0 & (16\sigma_i - 8)/3 & 2\pi & 2 & 0 \\ 1 & 0 & 2 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 2\pi\sigma_i \\ 8\sigma_i \\ 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} a \\ b \\ c \\ \lambda_1 \\ \lambda_2 \end{bmatrix}. \qquad (3.16)$$

Given σ_d and σ_i, (3.10) and (3.16) can be solved to completely specify a decimation and interpolation filter respectively. $\sigma_d, \sigma_i = 0$ gives filters with good[3] stopband performances, while $\sigma_d, \sigma_i = 1$ gives filters with good passband performances. Varying the values of σ_d, σ_i between these extremes causes a smooth variation in the frequency responses of the corresponding filters, as shown in Figure 3.9 where one dimensional decimation and interpolation filter responses are shown for different σ_d and σ_i.

In summary, an image can be filtered by convolving it with a 5×5 pyramid filter. However, a problem arises in filtering near the edge of an image where pixel are "missing". These issues are referred to as filtering edge effects and are discussed in the next section.

3.5.2 Filtering Edge Effects

Filtering edge effects can be handled in various ways, three of which are discussed below. These different techniques all substitute some value for that of an unknown pixel outside the borders of the image.

[3] A good filter response is close to the ideal response in a squared error sense.

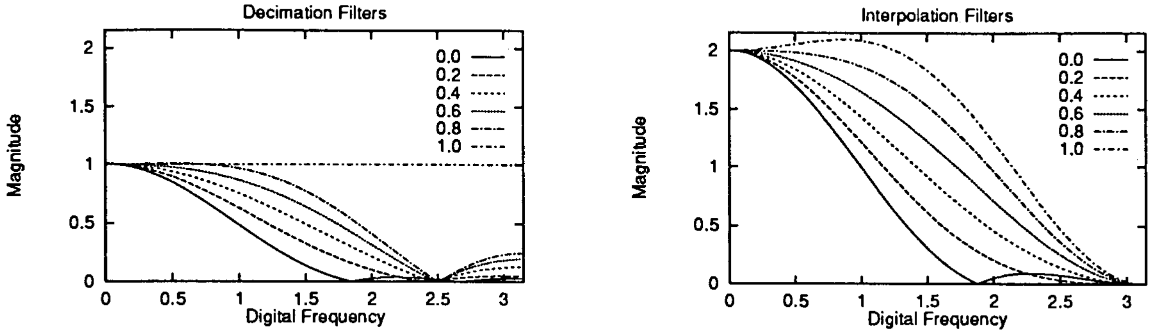


Figure 3.9: Decimation and Interpolation Filter Responses for Various σ_d and σ_i

Padded Edges

In this technique, a constant is substituted for unknown pixels. The constant is usually chosen to have some value close to the mean of the image being filtered. It is the simplest edge handling technique to implement, and is widely applied in practice.

Wrap-Around Edges

In this method, the image surface is extended by simulated joining of the image at its top and bottom, and left and right. This technique is analogous to circular convolution (Oppenheim and Schafer 1989).

Reflected Edges

Lastly, the image surface can be extended at its borders as if it were flipped and joined left, right, up, down and diagonally so that pixels on the immediate border of the original image would be duplicated across the original image boundary on the extended image surface. A similar edge handling technique has been previously applied in subband coding (Smith and Eddins 1987) and is referred to as symmetric extension. This technique has been found in practice to result in the lowest energy pyramid subimages, leading to the most efficient coding, and is therefore the edge handling method used throughout this research.

3.6 Pyramid Quantizers

3.6.1 Properties

Uniform quantization followed by entropy coding is asymptotically optimal with increasing bit rate (see Section 2.3.2), and has therefore been used in this research to quantize the pyramid subimages.

In this research, uniform quantizers are designed to be symmetrical about the mean of the image to be quantized (see Section 2.3.2). Since these quantizers have been constrained to give quantizer indices (see Figure 2.2) with integer bit rates, they must have 2^n quantization cells for some integer $n \geq 0$. For $n = 0$, the quantizer has a single quantization cell, so that all pixel values are quantized to the image mean. Whereas, for $n > 0$, a quantizer has an even number of cells and will therefore be midrise with a quantization boundary on the image mean, as opposed to being midtread with a quantization output on the image mean (see Figure 2.3). It should be noted here that midtread quantizers have the advantage of a "dead-zone" that, for highly non-uniform distributions, may result in a significant reduction in the entropy of the quantization indices. A complete uniform quantizer may therefore be defined with only four parameters, namely the rate (n), the image mean, the image standard deviation and the load fraction.

3.6.2 Design

Before designing a quantizer for a pyramid subimage, the rate of the quantizer must be specified. This is generally done with the use of a bit allocation scheme (see Section 2.3.2). The image mean and standard deviation are calculated as in equations (A.7) and (A.5). The optimal load fraction, in terms of maximizing the PSNR (A.13) of the quantized image, depends on the pixel value distribution of that image. The quantization process may be modeled as the addition of noise, or distortion, to the original image. Two types of distortion arise in the use of a quantizer, namely granular and overload distortion. Granular distortion represents the quantization noise added in the quantization of pixel values within the granular cells. Overload distortion on the other hand results from quantization of pixel values in the overload cells (see Figure 2.4). The choice of a load fraction below the optimal value leads to excessive granular distortion, while the choice of a load fraction above the optimal value leads to excessive overload distortion. The optimal load fraction in terms of minimizing the overall quantizer distortion therefore represents a tradeoff between granular and overload distortion. Figure 3.10 shows a typical graph of the PSNR of a quantized natural image as a function of the load fraction. This optimal load fraction can be calculated for a given image and rate using the iterative Golden Section algorithm (Press 1992), which minimizes the quantizer distortion in a MSE (A.10) sense with respect to the load fraction. Note that, for a given load fraction in this research, the distortion resulting from the quantization of an image was measured by first quantizing the image, and then measuring the resulting distortion. However, in a practical application, this distortion could be estimated at lower computational cost based on prior knowledge of the image variance and pixel pdf.

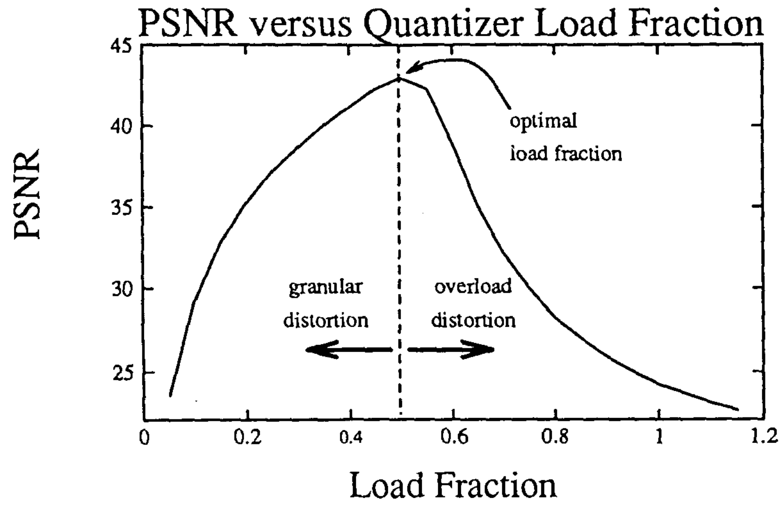


Figure 3.10: Quantizer Distortion as a Function of Load Fraction

3.6.3 Quantization Noise in the Pyramid

Quantization noise from the pyramid subimage quantizers can be approximated as white (Gersho and Gray 1992), with the accuracy of this approximation improving with increasing quantizer rate. However, the spectrum of the overall noise introduced into the reconstructed image, indirectly from the quantization of the pyramid subimages, depends on the pyramid generation scheme (Uz, Vetterli, and LeGall 1991). For a pyramid generation without quantization feedback (see Figure 3.3), quantization noise accumulates at low spatial frequencies during the image reconstruction process. This is due to the lowpass filtering done in the interpolation process, which attenuates mainly the high spatial frequency quantization noise prior to adding the interpolated image to the next lower level of the pyramid. Since the quantization noise in the interpolated subimage and the image to which it is being added are uncorrelated, they will add arithmetically, resulting in an overall quantization noise spectrum that is biased towards low spatial frequencies. This is particularly undesirable, since the HVS is most sensitive to low spatial frequencies. However, for a pyramid generated with quantization feedback (see Figure 3.4), quantization noise is corrected at each stage of the interpolation during the image reconstruction process. Therefore, quantization noise in the reconstructed image results only from the quantizer used for the largest subimage, or pyramid base. This quantization noise is approximately white and less offensive to the HVS. The latter pyramid generation scheme is therefore more desirable from a quantization noise point of view.

3.7 Codeword Assignment in the Pyramid

In order to exploit statistical redundancy present in the quantization indices, it is desirable to apply some form of entropy coding in the codeword assignment process. Huffman coding (Section 2.3.3) is an entropy coding technique that assigns integer length codewords uniquely to each quantizer index to achieve a low overall rate. For the images used in this research, the overall rate achieved through Huffman coding has been demonstrated to be close to the optimal lower entropy bound. In addition to this, Huffman codes are simply designed, efficiently implemented in practice, and have the desirable property of instant decodability. For these reasons, Huffman coding has been used for all entropy coding in the simulations performed in this research.

However, when codeword assignment must be done for a very low entropy distribution, Huffman coding may not result in a rate close to the entropy lower bound. This is due to the constraint that each Huffman codeword must be of length greater than or equal to one bit. Consequently, the overall Huffman code rate will be greater than or equal to one bit per symbol. For such low entropy distributions, or where there exists appreciable intersymbol correlation, there may be significant advantage in using an alternative entropy coding technique, for example arithmetic coding (Section 2.3.3).

Chapter 4

Pyramid Coding Of Still Images

As previously discussed, pyramid coding can be tailored to suit a particular application. Lossless or lossy image coding, or the coding of images for progressive transmission are examples of such applications. Some of the issues that arise in the use of pyramid coding in these different applications will now be discussed.

4.1 Lossless Coding

In lossless image coding, the objective is to represent an image using the least number of bits without introducing any distortion whatsoever into the reconstructed image. This objective can be achieved by minimizing the statistical redundancy generally present in the original image representation (see Chapter 2). Minimization of the statistical redundancy can be achieved by minimizing the entropy (A.8) of the image representation. This in turn maximizes the compression ratio (A.14). The first lossless pyramid coding scheme proposed for images (Wang and Goldberg 1989a) is outlined in Figure 3.4, but is not practical in real lossless coding applications due to the fact that it does not allow quantization of the base subimage of the pyramid without distortion being introduced into the reconstructed image. Subsequently, a more practical lossless pyramid coding scheme was proposed (Goldberg and Wang 1991) in which the pyramid subimages were constrained to have integer pixel values. In general, lossless pyramid coding can be applied to images as outlined in Figure 3.5.

Some important issues arise in the application of lossless pyramid coding to images. As previously described, the choice of pyramid filters used in the generation scheme has a significant influence on the properties of the generated pyramid. In order to generate pyramids suitable for lossless coding, the entropy of the pyramid must be minimized. Here, the entropy of a pyramid is defined as the combined entropy of all of the pyramid subimages together, and is a straightforward extension of (A.8). As outlined in Chapter 3, the base of the pyramid is the same size as the original image. Since a factor of 2 is used in the subsampling and upsampling processes, each subsequent pyramid level generated will have 1/4 the number of pixels of its predecessor. Therefore, the factor R by which pyramid coding increases the total number of pixels, when compared to the original image, may be represented by the sequence

$$R = \sum_{i=0}^{N-1} \frac{1}{2^{2i}} = \frac{4}{3}(1 - 2^{-2N}), \tag{4.1}$$

where N is the number of pyramid levels and $i = 0$ for the pyramid base subimage. In the limit, N tends to infinity and it is easily shown that this ratio tends to 4/3. This is an undesirable property of pyramid coding, since increasing the number of pixels in an image representation increases the number of bits needed to send or store the image, and therefore decreases the compression ratio that can be achieved. Pyramid coding also requires the convolution of images with decimation and interpolation filters, a computationally expensive process even for the new class of 5×5 FIR pyramid filters developed in Section 3.5.1.

In the next section, a novel implementation of pyramid coding, suitable for lossless image coding, will be developed. This technique not only achieves a low pyramid entropy through appropriate choice of pyramid filters, but also allows an image to be represented in a subsampled pyramid of the same number of pixels, and can be efficiently implemented at low computational cost.

4.1.1 Minimal Entropy Pyramid Coding

Minimal entropy pyramid (MEP) coding (Houlding and Vaisey 1994) requires the use of the pyramid coding scheme outlined in Figure 3.5, in which the filters defined in Section 3.5.1 are used in the decimation and interpolation processes, and quantization of the filter outputs is to the nearest integer.

A search was done to determine which pyramid filter pair, within the developed filter class, generates a pyramid with minimal entropy. Six level pyramids were generated using various different decimation and interpolation filters from the new filter classes. In Figure 4.1, the pyramid entropy is shown as a function of σ_d and σ_i (see equations 3.10 and 3.16), for "Lenna" in Figure B.1. For a wide variety of test images, the following decimation filter, $g(\sigma_d, i, j)$, and interpolation filter, $h(\sigma_i, i, j)$, were found to be optimal, within the new filter class, in terms of minimizing the pyramid entropy

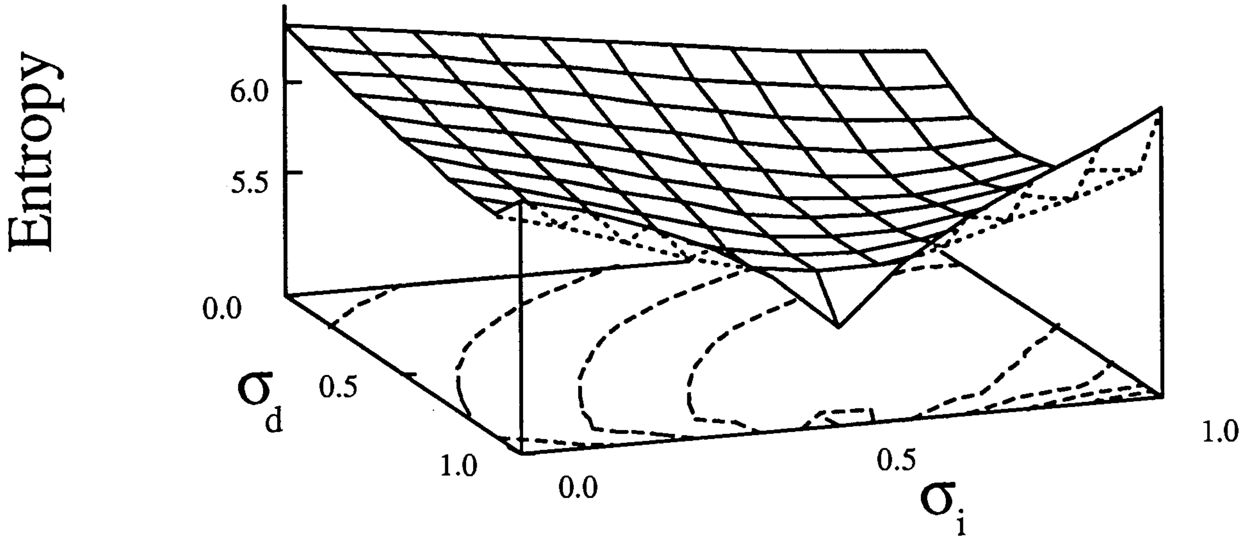


Figure 4.1: Lenna$_l$ Pyramid Entropy versus Decimation and Interpolation Filter Type

$$g(1.0, i, j) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$h(0.5, i, j) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1/4 & 1/2 & 1/4 & 0 \\ 0 & 1/2 & 1 & 1/2 & 0 \\ 0 & 1/4 & 1/2 & 1/4 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{4.2}$$

The pyramid generated using the filters in (4.2) shall be referred to as the *minimal* *entropy pyramid* (MEP). It is interesting to note that the effective size of the interpolation filter is 3×3 , and that the "center" coefficient is unity. This property, together with the fact that the decimation filter is a unit impulse, ensures that pixels subsampled from the original image in the decimation process are interpolated to their original values in the interpolation process. Subtraction then forces 1/4 of the pixels in all of the pyramid subimages except the smallest to zero, causing the sharp entropy minima in Figure 4.1, and allows the MEP to be subsampled so that a given image may be represented in an equivalent MEP of the same number of pixels. This is in contrast with pyramid coding in general, for which the number of pixels to be coded increases by a factor of approximately 4/3.

It is interesting to note that the decimation and interpolation filter coefficients in (4.2) may all be represented as powers of 2. This allows the decimation and interpolation filtering operations to be closely approximated using only integer additions and bit shift operations at a considerable saving in computational cost. Through this use of only integer operations, the MEP is guaranteed to contain only integers, and no quantization is necessary in the decimation and interpolation processes, leading to a further reduction in computational cost.

Given an integer image with n bit precision, every pixel value exists in the integer finite field, or Galois Field, $GF(2^n)$ (Lin and Costello 1983). Both the MEP decimation and interpolation processes are approximated by integer operations that produce pixel values also in $GF(2^n)$. However, subtraction of the interpolated image from the original image produces integer pixel values in the finite field $[-2^n + 1, 2^n - 1]$. This is undesirable, since it means that an extra bit is needed to represent each of the pixels in the difference image. However, since the pixels in both the original and interpolated images exist in $GF(2^n)$, it is also possible to represent the difference image pixels in $GF(2^n)$. This requires that the difference image be computed using $GF(2^n)$ finite field arithmetic. Through this use of finite field arithmetic in MEP coding and decoding, the MEP pixels are assigned intensities on $GF(2^n)$, and may be represented in n bits. This is an extension of the use of finite field arithmetic previously applied in a another lossless hierarchical coding method (Torbey and Meadows 1989).

There are at least two related hierarchical coding schemes that are in the same class as the MEP coding scheme previously outlined. These are namely *Reduced Difference Pyramid* (RDP) coding and *Paired Pyramid* (PP) coding. To facilitate later comparison, these alternative schemes will now be outlined briefly.

4.1.2 Reduced Difference Pyramid Coding

RDP coding was first proposed as a lossless image coding technique by Wang and Goldberg (1989b), and later evaluated against related techniques (Goldberg and Wang 1991), where it was found achieve the highest performance in terms of three criteria : the equivalent entropy, the rate-distortion performance in progressive transmission and total lossless transmission bit rate.

The RDP is based on the truncated mean pyramid. To illustrate the derivation of the truncated mean pyramid, consider a simple 2×2 image with integer valued pixels x_i as shown in Figure 4.2. Parent node y_0 is calculated as

$$y_0 = Q(\frac{x_0 + x_1 + x_2 + x_3}{4}) \tag{4.3}$$

where $Q()$ denotes quantization to the nearest integer. This basic iteration is easily extended for larger images, where each 2×2 group of nodes in the original image produces one parent node. A two level pyramid is generated on the first pass, which may then be repeated on the resulting image of parent nodes to give a three level pyramid and so forth. A high degree of spatial correlation generally exists in the various levels of the truncated mean pyramid, leading to statistical redundancy. The RDP reduces this redundancy by representing the pixels x_{0-3} as y_{0-3}, where

$$\begin{aligned} y_0 &= \text{as above}, \\ y_1 &= x_1 - x_3, \\ y_2 &= x_2 - x_0, \\ y_3 &= x_3 - x_2. \end{aligned} \tag{4.4}$$

The differencing used in calculation of the RDP effectively decorrelates the pixels x_{0-3}. Note that only three difference coefficients y_{1-3} plus the parent coefficient y_0 are needed to represent the original pixels x_{0-3}. Therefore, an image can be represented in a RDP of the same number of "pixels". This iteration may be applied in a reversible manner, starting at the base and moving up the truncated mean pyramid. It is easily shown that, using the iteration below, the original pixels x_{0-3} can be recovered exactly in an efficient stepwise fashion.

$$\begin{aligned} x_1 &= y_0 + Q(\frac{3y_1 + 2y_3 + y_2}{4}), \\ x_3 &= x_1 - y_1, \\ x_2 &= x_3 - y_3, \\ x_0 &= x_2 - y_2. \end{aligned} \tag{4.5}$$

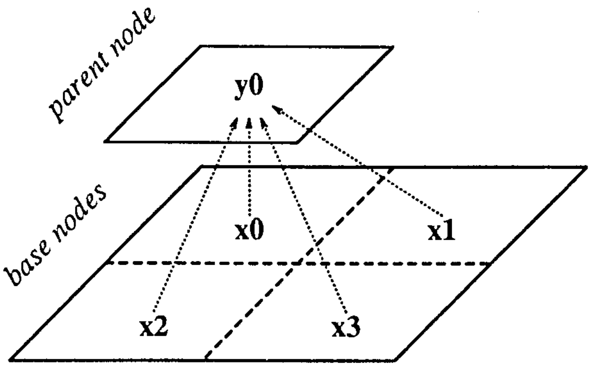


Figure 4.2: A Simple 2×2 Image

The truncated mean pyramid can be thought of as a Gaussian pyramid created using a 2×2 rectangular mean filter. However, the RDP differs significantly from the Laplacian pyramid, and therefore the MEP. This is due to the different manner in which the RDP is computed.

4.1.3 Paired Pyramid Coding

PP coding, another technique suitable for lossless image coding, was first proposed by Torbey and Meadows (1989). The basic PP coding iteration is illustrated in Figure 4.3. This iteration is applied to neighboring pixels in a row or column of the image to be coded. For example, neighboring pixels x_0 and x_1 in an n bit, $\mathrm{N} \times \mathrm{M}$ dimensional image may be coded as follows. One of the pixels, x_0, is carried through the coding iteration unaffected while the other, y_0, is calculated as

$$y_0 = (x_1 - x_0)\%2^n, \tag{4.6}$$

where "%" denotes the modulo operation. If this coding iteration is initially applied to each pair of neighboring pixels in the image rows, two $\mathrm{N} \times \frac{M}{2}$ images are produced. The first subimage is a version of the original that has been subsampled by a factor of two horizontally and contains the x_0 coefficients, while the second subimage contains the modulo difference coefficients y_0. The coding iteration is then applied to pairs of neighboring pixels in the columns of the $\mathrm{N} \times \frac{M}{2}$ subimage of x_0 coefficients to produce two $\frac{N}{2} \times \frac{M}{2}$ subimages and so forth. This alternating application of the row and column iterations successively on the subimage of x_0 coefficients is continued until at least one of the subimage dimensions is reduced to one. The decoding operation proceeds in the reverse fashion from the smallest PP subimage and recovers the original neighboring pixel pairs x_0 and x_1 as follows. x_0 is carried through the decoding iteration unaffected, while x_1 is calculated as

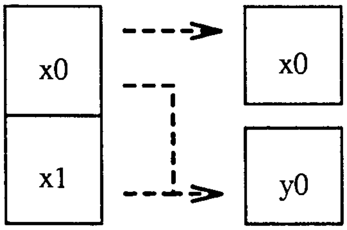


Figure 4.3: Paired Pyramid Coding Iteration

$$x_1 = (y_0 + x_0)\%2^n. \tag{4.7}$$

The modulo differencing operation has the effect of decorrelating the pixels in the original image and leads to a concentration of y_0 values around 0 and $2^n - 1$. This in turn generally leads to a significant reduction in the entropy of the original image and allows efficient lossless compression. The modulo operation also allows the difference coefficients to be represented in the same number of bits as the pixels in the original image. Clearly, an image can be represented in the form of a paired pyramid of the same number of pixels.

4.1.4 Simulations

MEP coding can be considered to be a lossless transformation that generally results in a lower entropy representation of an image. Since the entropy of an image depends on its pixel value distribution, it will be useful to examine the pixel value distributions of a test image and its MEP equivalent representation. The image "Lenna" in Figure B.1 was used to generate a ten level MEP of the same number of pixels. The pixel value distributions of "Lenna" and the equivalent MEP are shown in Figure 4.4. Clearly, the test image has a well distributed range of pixel values, while the MEP has a "peaky" distribution with pixel values concentrated around 0 and 255. Note that the concentration of pixel values around 255 results from the use of modulo arithmetic in the MEP coding operation. These values would otherwise be negative, and concentrated around 0.

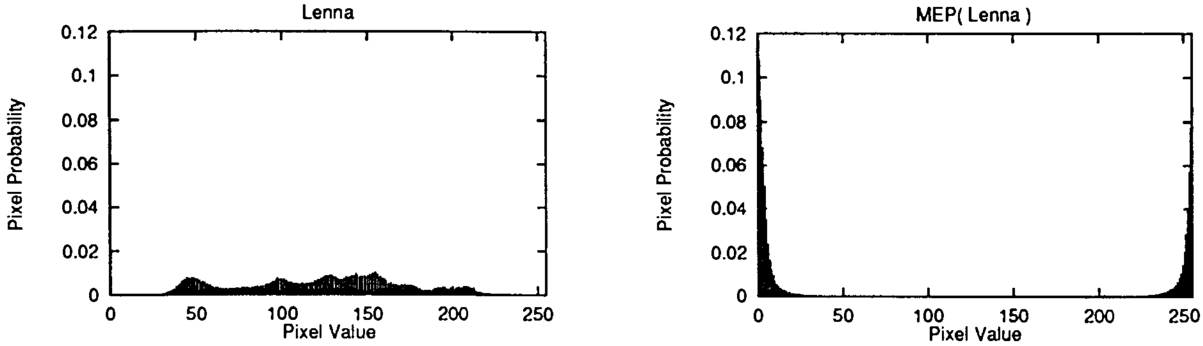


Figure 4.4: Pixel Value Distributions of Lenna and MEP Equivalent

In order to evaluate the performance of MEP coding in terms of compression ratios, simulations were done in which a set of test images were compressed using both MEP coding and a DPCM scheme. In the DPCM scheme, a given pixel was predicted using the three known neighboring pixels, plus an added constant. Predictor coefficients were calculated based on the covariance matrix of the test image being compressed. This DPCM scheme was chosen for the comparison since it is an alternative lossless coding scheme that, like MEP coding, exploits two dimensional interpixel correlation in order to achieve a good compression. DPCM is a well known and widely applied technique (Gersho and Gray 1992).

In summary, both the MEP and the DPCM residual error image were generated for a given test image, after which they were Huffman coded for lossless compression. The compression ratio was then calculated as the ratio of the test image file size to the Huffman coded file size. The compression ratios did not include the bits needed to send the Huffman code table or predictor coefficients, but this does not significantly effect the results presented. Test images are shown in Figure B.1. Lenna$_d$ was obtained by first filtering Lenna with a 32 tap QMF (Jayant and Noll 1984), and then subsampling by a factor of two horizontally and vertically. The results of these simulations are summarized in Table 4.1.

Table 4.1: Compression Ratio Performance of Proposed MEP System

Image			Entropy		Compression Ratio	
Name	Width	Height	MEP	DPCM	MEP	DPCM
Boat	512	512	4.663	4.444	1.71	1.79
Lenna	512	512	4.595	4.526	1.73	1.75
Lenna$_d$	256	256	4.948	4.822	1.61	1.65
Mandrill	512	512	6.379	6.208	1.25	1.28
Peppers	512	512	5.056	5.215	1.58	1.52

It can be seen that MEP coding has comparable performance to the DPCM scheme. In simulations performed using a much larger set of test images, DPCM was found to give a marginal average performance improvement in compression ratio of 1.8%. However, in a real application, the computational cost involved in the two coding schemes would be an important consideration. MEP coding requires only integer operations, while DPCM requires many costly floating point operations, including multiplications. DPCM also requires an additional image prescan to calculate predictor coefficients. In the case where predictor coefficients are not calculated separately for each image being coded, it is expected that the performance of the DPCM scheme will become inferior that of the MEP scheme. It is also possible to apply the DPCM scheme to the MEP subimages separately. With DPCM coding of only the largest MEP subimage, or base of the pyramid, a 0.25% marginal average performance improvement of MEP over DPCM coding was observed for the same set of test images. However, in this case, the computational advantage of MEP coding is lost.

For natural images in general, the entropy of the MEP is high enough that Huffman coding can achieve a good efficiency, resulting in a rate, in bits per pixel, close to the entropy lower bound. However, for a MEP with a significantly lower entropy, resulting from an image with higher interpixel correlation, Huffman coding becomes inefficient due to the fact that codeword lengths are constrained to be integers greater than zero. In this case, an alternative lossless coding technique could be used after MEP coding, for example arithmetic coding (Bell, Cleary, and Witten 1990).

The MEP coding results represent significant improvements over similar lossless compression results for 8 bpp originals. Table 4.2 summarizes the results of related research in comparison to MEP coding.

Table 4.2: Relative Performance of Proposed MEP System

Research	Method	Image	Entropy	
			Result	MEP
Wang and Goldberg (1989a)	pyramid coding and vector quantization	Boat	5.862	4.663
Furlan (1991)	optimal model based arithmetic coding	Lenna$_d$	5.100	4.948
Ho and Gersho (1989a)	transform coding vector quantization	Lenna Boat	5.067 5.262	4.595 4.663
Wang and Goldberg (1989b)	reduced difference pyramid coding	Lenna Boat	5.084 4.955	4.595 4.663
Torbey and Meadows (1989)	paired pyramid coding	Lenna Boat	5.040 4.992	4.595 4.663

4.2 Lossy Coding

In efficient lossy coding, the objective is to achieve the highest possible image quality at a given rate. Since there is no requirement for perfect reconstruction of the original image, coding can take advantage of statistical as well as perceptual redundancy generally present in natural images. Lossy coding therefore typically achieves higher compression ratios than lossless coding.

Lossy pyramid coding of images consists of the following steps. Firstly, a pyramid is generated from the image. Next, bits must be allocated to each of the subimages in the pyramid. Quantizers may then be designed based on these bit allocations, and each subimage quantized. Finally, the quantized subimages may be entropy coded to produce a pyramid representation suitable for storage or transmission.

4.2.1 Pyramid Generation

Either of the schemes outlined in Figures 3.3 or 3.4 can be used for pyramid generation. As mentioned in Section 3.6.3, the former scheme does not use quantization feedback, while the latter does. Quantization feedback is desirable from the point of view that it allows more control of the distortion introduced indirectly into the reconstructed image through quantization of the pyramid subimages. The pyramid generation scheme outlined in Figure 3.4 was therefore chosen for lossy pyramid coding. The decimation and interpolation filters may be chosen from the developed class of pyramid filters (see Section 3.5.1) through the specification of σ_d and σ_i respectively. For a given natural image, there generally exists a unique optimal pyramid filter pair that can be used to generate a pyramid best suited to efficient lossy coding. Some of the considerations involved in the choice of decimation and interpolation filters for generation of pyramids suitable for lossy coding will now be discussed.

A value σ_d that is too low specifies a decimation filter that is too lowpass and does not allow sufficient information from the original image to propagate through the decimation process to the pyramid top. More information from the original image must therefore be represented in the larger pyramid base. This leads to a higher variance pyramid base, resulting in inefficient coding. On the other hand, a value of σ_d that is too high specifies a decimation filter that is not sufficiently lowpass and therefore does not adequately attenuate the high spatial frequencies in the original image prior to subsampling. This leads to significant aliasing in the decimation process, causing an increase in variance for both the pyramid base and top with no corresponding increase in the quality of the reconstructed image, and therefore results in inefficient coding.

Similarly, a value of σ_i that is too low specifies an interpolation filter that is too lowpass and therefore does not allow sufficient information to propagate back from the pyramid top to the interpolated image in the interpolation process feedback loop. This results in a higher variance for the larger pyramid base and therefore inefficient coding. On the other hand, a value of σ_i that is too high specifies an interpolation filter that is not sufficiently lowpass and does not adequately attenuate the unwanted "interpolation image", an undesirable byproduct of the upsampling operation. This leads to "false" information being propagated to the interpolated image, causing an increase in the variance of the pyramid base and inefficient coding.

Bit allocation must be done to determine the rate of each subimage, allowing the design of the corresponding optimal uniform quantizers as described in Section 3.6.2. After quantization, the subimages may be entropy coded for storage or transmission (see Section 3.7).

4.2.2 Bit Allocation

If each subimage is regarded as a source of image information contributing to the quality of the overall reconstructed image, either the BFOS or the Greedy algorithm can be used for bit allocation (see Section 2.3.2). In order to reconstruct the best possible approximation to the original image from the quantized pyramid subimages, it is desirable to allocate more bits (a higher rate) to the more perceptually important subimages. In this research, the MSE (A.10) has been used to evaluate the overall distortion of a given reconstructed image, when compared to the original image from which the pyramid was generated. Some difficulties exist in the implementation of these bit allocation schemes in a lossy pyramid coding application. To avoid these complications, it is possible to do an exhaustive search for the optimal allocation at a given rate. Given a pyramid, this would be done by trying every possible bit allocation, quantizing the subimages accordingly for each allocation, reconstructing the image from the quantized pyramid, and measuring the overall rate and distortion. The allocation with an overall rate less than or equal to the target rate that results in the lowest overall distortion would then be chosen as optimal. However, this approach is extremely computationally expensive and becomes impractical for all but the smallest images and sets of possible bit allocations. At least two alternative, more practical solutions to this problem exist.

In the first approach, an approximate optimal bit allocation for a pyramid with a given overall rate can be found experimentally, given a set of test images that are a representative subset of the images to be coded later using lossy pyramids. Alternatively, some modifications to the BFOS or Greedy algorithms can be made to facilitate their use in bit allocation for lossy pyramid coding. These modifications will now be discussed briefly.

BFOS Bit Allocation

In the BFOS algorithm (Riskin 1991), the overall rate R for N sources is calculated as

$$R = \sum_{i=0}^{N-1} p_i \, r_i, \tag{4.8}$$

where p_i and r_i represent the probability and rate of source i respectively. Similarly, the overall distortion D is calculated as

$$D = \sum_{i=0}^{N-1} p_i \; d_i, \tag{4.9}$$

where d_i represents the MSE distortion resulting from quantization of source i with an r_i bit quantizer. The BFOS algorithm determines the optimal bit allocation $\{r_i : i = 0, \cdots, N-1\}$ with an overall rate less than or equal to some target rate. It should be noted however that this allocation is optimal only in the sense that it is guaranteed to lie on the convex hull of the rate-distortion function, and therefore result in the lowest distortion D at that rate R. In some cases, it may be possible to allocate more bits without exceeding the target rate using an alternative allocation scheme. In pyramid coding, where for the purpose of bit allocation each subimage is considered to be a source, there are some complications in implementation of the BFOS algorithm in this form. These will be discussed below, along with proposals for modification of the BFOS algorithm that enable it to cope with the peculiarities of bit allocation in pyramid coding.

A similar linear combination of subimage rates to that in (4.1) can be used for calculation of the overall rate R of an N level pyramid. In this case, the weighting coefficients are determined from the ratio in sizes of the pyramid subimages and the original image. Clearly, these coefficients do not sum up to one, and cannot be equated to source probabilities as in (4.8). To overcome this problem, the probabilities p_i can be set to some arbitrary value, and the subimage rates r_i preweighted before use in the BFOS algorithm. Let $p_i = \frac{1}{N}$ for all i. The preweighted r_i, denoted r_i', may then be calculated as

$$\begin{aligned} p_i \; r_i' &= \frac{1}{2^{2i}} \; r_i, \\ r_i' &= \frac{N}{2^{2i}} \; r_i, \end{aligned} \tag{4.10}$$

allowing calculation of the overall rate R of the pyramid as in (4.8), with r_i' substituted for r_i. Note that $i = 0$ for the pyramid base.

The overall distortion in lossy pyramid coding is calculated as the MSE of the image reconstructed from the quantized subimages, when compared to the original image. Calculation of the overall distortion as a simple linear combination of the distortions in each pyramid subimage poses two difficulties. Firstly, quantization noise in all subimages other than the base of the pyramid gets interpolated during image reconstruction. The effect of the upsampling and interpolation filtering operations must therefore be incorporated into calculation of the overall distortion. Secondly, in the case of a pyramid generated with quantization feedback, quantization noise in these subimages is first interpolated and then requantized. Since quantization is a nonlinear problem, the estimation of overall distortion in the reconstructed image from the quantization noise in each subimage becomes a complicated problem. However, with some simplifying assumptions, it is possible to estimate the overall distortion as a linear combination of the separate subimage distortions. Based on the assumption that interpolated quantization noise has significantly lower energy than the subimage to which it is being added, its effect on the quantization noise at that level of the pyramid may be ignored. The effect of quantization feedback in the pyramid generation process is thereby ignored. On the other hand, the effect of the interpolation process may be incorporated into calculation of each of the subimage distortion weighting coefficients. A derivation will now be presented for calculation of these distortion weighting coefficients.

To simplify this derivation, consider a one dimensional white noise vector N_{in} with variance σ^2_{in}. This noise is passed through the interpolation process as shown in Figure 4.5 where $G(z)$ is the lowpass interpolation filter. We strive to find σ^2_{out}, the variance of the interpolated noise N_{out}. This leads to the factor β by which the noise variance changes through the interpolation process, where

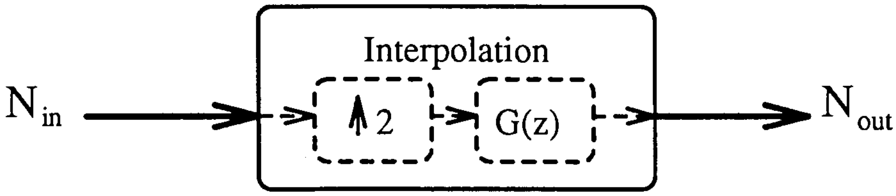


Figure 4.5: Interpolation of Noise

$$\beta = \frac{\sigma^2_{out}}{\sigma^2_{in}}. \tag{4.11}$$

In the context of subband coding, Woods and Naveen (1992) have shown that β may be calculated as

$$\beta = \frac{1}{2}\sum_{-\infty}^{\infty} |g(n)|^2, \tag{4.12}$$

where $g(n)$ is the impulse response of the one dimensional interpolation filter (3.3) defined by σ_i and (3.16). For a two dimensional separable interpolation filter with identical horizontal and vertical component filters, β is squared. Therefore, the overall distortion D for an N level pyramid would calculated as

$$D = \sum_{i=0}^{N-1} \beta^{2i} \; d_i, \tag{4.13}$$

where β is calculated as in (4.12) and d_i is the MSE distortion resulting from quantization of subimage i. A strategy similar to that defined in (4.10) can then be used for calculation of the preweighted distortion coefficients d_i' as

$$d_i' = N \; \beta^{2i} \; d_i, \tag{4.14}$$

allowing the overall distortion D to be estimated as in 4.9, with d_i' substituted for d_i. Given the probabilities $p_i = \frac{1}{N}$ and the preweighted rates r_i' and distortions d_i', the BFOS algorithm may be used to allocate bits to each subimage i of the pyramid.

To evaluate BFOS bit allocation in lossy pyramid coding, a simulation was done in which the BFOS allocations were compared against all possible allocations. The image for this simulation was created by decimating "Boat" (see Figure B.1) three times using a decimation filter created with $\sigma_d = 0.0$. This image was then used to generate an $N = 3$ level pyramid. All possible allocations $\{0 \leq r_i \leq 8 : i = 0, \cdots, N-1\}$ were made in turn to the subimages of the pyramid. For a given allocation, the subimages were quantized, starting at the pyramid top. After quantization of a given subimage, the quantization error was fed back to the next lower level of the pyramid prior to design of the next quantizer, and so forth (see Figure 3.4). For each allocation $\{r_i : i = 0, \cdots, N-1\}$, the overall rate R was calculated as in 4.1, while the overall distortion D was measured as the MSE distortion (A.10) of the image reconstructed from the quantized pyramid, when compared to the original image. The input data to the BFOS algorithm was calculated as previously discussed, and the allocations generated for a given target rate. Table 4.3 shows the set of optimal allocations calculated using the BFOS algorithm, where $\hat{R}$, R, $\hat{D}$ and D represent the allocated and target rate, and estimated and actual distortion respectively. As can be seen, the estimated overall distortion $\hat{D}$ is close to the actual overall distortion D, except at low allocated rates $\hat{R}$, where the assumption that subimage quantization noise is white becomes inaccurate and the effects of requantization significant. It is also interesting to note

Table 4.3: Optimal BFOS Bit Allocations for Boat (64 × 64)

Allocation			Rate (bpp)		MSE	
r_0	r_1	r_2	$\hat{R}$	R	$\hat{D}$	D
1	1	2	1.375	2.000	159.546	218.052
2	2	3	2.688	3.000	59.097	54.706
3	3	4	4.000	4.000	18.021	14.968
4	3	4	5.000	5.000	7.019	4.186
4	4	5	5.312	6.000	5.082	4.334
5	5	6	6.625	7.000	1.428	1.200
6	6	7	7.938	8.000	0.384	0.336

that D is generally less than $\hat{D}$. This is attributed to the use of quantization feedback in the pyramid codec. Figure 4.6 shows the rate-distortion performance for all possible allocations against the optimal allocations calculated using the BFOS algorithm. Note the logarithmic decrease in distortion with increasing rate, measured in bits. The horizontal tails of rate-distortion points result from inappropriate allocation of bits to subimages, resulting in an increase in the overall rate of the pyramid, with no corresponding decrease in overall distortion. Ideally, the allocated points should give the lowest distortion for a given rate. While the performance of the BFOS algorithm is reasonable at low rates, it is far from optimal at higher rates. This suboptimal performance was observed in similar simulations on various other test images, and is attributed to inaccuracies in $\hat{D}$, the estimated overall distortion.

Alternatively, the Greedy algorithm may be used for bit allocation to the subimages of the pyramid.

Greedy Bit Allocation

In the Greedy algorithm for bit allocation (Gersho and Gray 1992), bits are allocated to a set of sources according to their demand. Variations of the Greedy algorithm exist, based on how the demand is calculated. In applying Greedy bit allocation to pyramid coding, each subimage is again considered to be a source. A proposal will now be made for a Greedy bit allocation scheme suited to lossy pyramid coding. The basic algorithm for an N level pyramid is broken down as follows.

1. Initialize subimage allocations $\{r_i = 0 : i = 0, \cdots, N-1\}$.

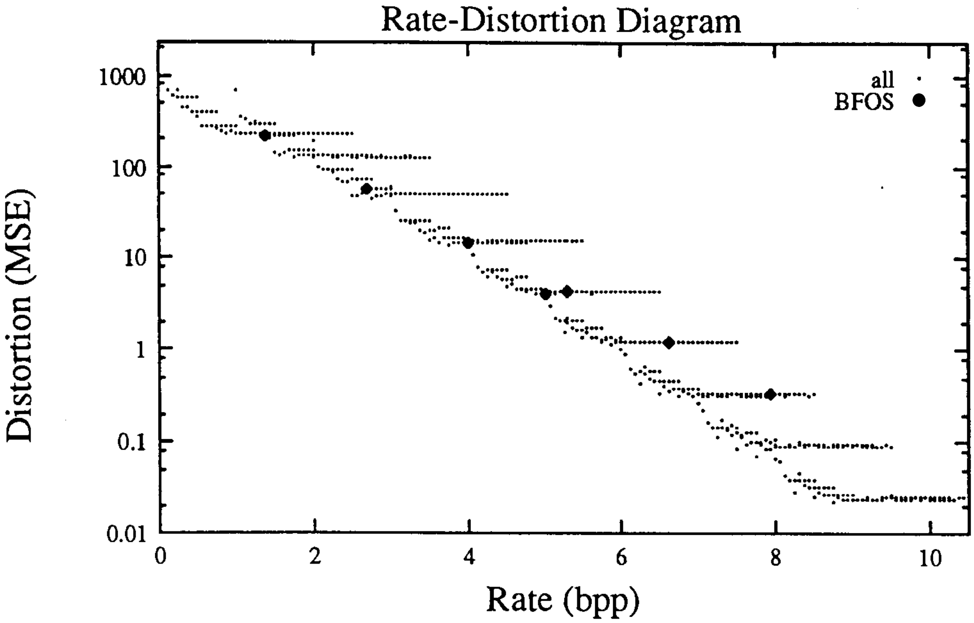


Figure 4.6: BFOS Rate-Distortion Performance for Boat 64 × 64

2. Find the source i with the maximum demand Δ_i.

3. Increment the bit allocation r_i for the source i with the maximum demand.

4. If the overall rate $\hat{R}$ is not sufficiently close to the target rate R, repeat from 2, otherwise stop.

The demand Δ_i of source i is defined as the drop in overall distortion D when one more bit is allocated to source i. Here, D is again measured as the MSE (A.10) of the image reconstructed from the quantized pyramid, when compared to the original image. The overall rate R may be calculated as in (4.1). In summary, for a given stage in the Greedy allocation, the current overall distortion is initially measured by quantizing the subimages based on the current allocation, reconstructing the image, and measuring D. The bit allocation r_i for each subimage i is then incremented in turn. For each increment, the overall distortion D_i is again measured as above. The demand for that subimage is then calculated as $\Delta_i = D - D_i$. The allocation for the subimage with the maximum demand is then incremented, and a check made to see if the overall pyramid rate R is sufficiently close to the target rate. If not, the procedure is repeated.

The performance of the Greedy algorithm is suboptimal in that it may not always select the allocation $\{r_i : i = 0, \cdots, N-1\}$ with the lowest overall distortion D at a given rate R. However, in practice its performance has proven to be sufficiently close to optimal for most lossy pyramid coding applications.

To evaluate the performance of the proposed Greedy bit allocation scheme in pyramid coding, some simulations were done in which the rate-distortion performance of the Greedy allocations were compared to all possible allocations for a given pyramid. An $N = 3$ level pyramid was initially generated from a 64×64 version of the standard test image "Boat" (see Figure B.1), as in the previous simulation done to evaluate the performance of the BFOS bit allocation scheme. For all possible allocations $\{0 \leq r_i \leq 8 : i = 0, \cdots, N-1\}$, the subimages were quantized. In each case, the image was reconstructed from the quantized pyramid, and the overall rate R and distortion D measured as previously described. Bit allocations were then made using the Greedy bit allocation scheme previously defined. The allocations r_i are shown for the three level pyramid in Table 4.4, where $i = 0$ for the pyramid base. $\hat{R}$, R and D represent the overall measured rate, the overall target rate, and the overall measured distortion respectively. Note that for a bit allocation $r_i = 0$ for subimage i, all pixels in subimage i are quantized to the subimage mean.

Table 4.4: Greedy Bit Allocations for Boat (64×64)

Allocation			Rate (bpp)		MSE
r_0	r_1	r_2	$\hat{R}$	R	D
0	1	0	0.250	1.000	683.813
1	1	0	1.250	2.000	302.732
2	1	0	2.250	3.000	85.303
3	1	0	3.250	4.000	24.708
4	1	0	4.250	5.000	6.376
5	1	0	5.250	6.000	1.566
6	1	0	6.250	7.000	0.433
7	1	0	7.250	8.000	0.111

The bit allocations in Table 4.4 are plotted against the full set of possible rate-distortion points for comparison in Figure 4.7. For all rates, the Greedy algorithm gives optimal or near optimal performance. Furthermore, with a slight modification,

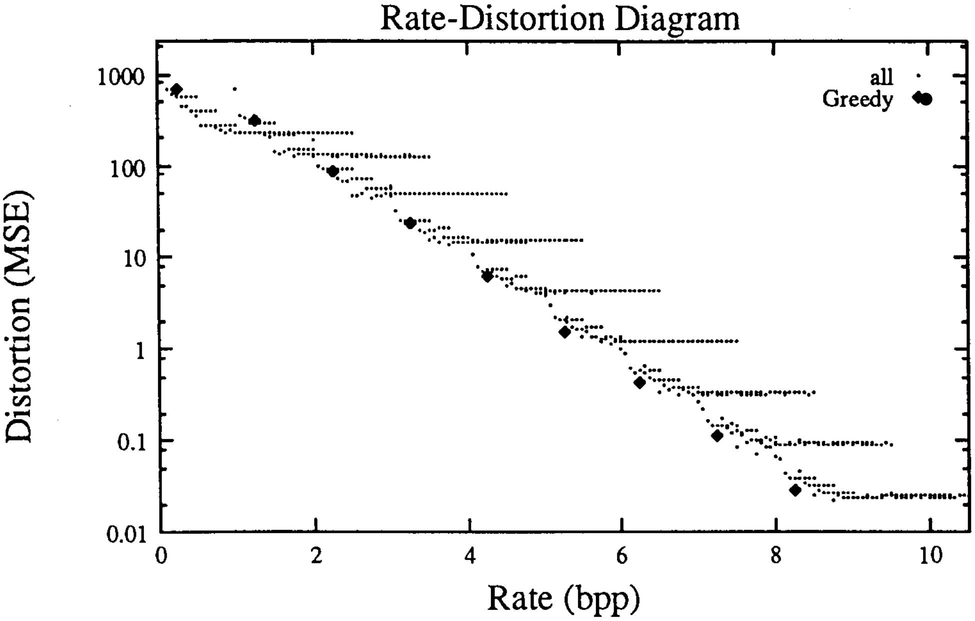


Figure 4.7: Greedy Rate-Distortion Performance for Boat 64 × 64

the Greedy algorithm defined above can be improved. As discussed, the Greedy algorithm allocates a bit in each iteration to the subimage with the greatest demand. When it cannot allocate a bit to the subimage with the greatest demand without exceeding the overall target rate R, bit allocation stops. For some target rates, although it is not possible to allocate more bits to the subimage with the greatest demand, it may be possible to allocate more bits to a subimage with less demand. This would allow additional bits to be allocated near the end of the Greedy allocation algorithm, further reducing the overall distortion D while remaining below the target rate R. In summary, the modified Greedy algorithm proceeds, as previously defined, until it cannot allocate a bit to the subimage with the greatest demand without exceeding R. It then proceeds to make a further allocation to the subimage with the next greatest demand that will not cause the overall rate $\hat{R}$ to exceed the target rate R. Therefore, modified Greedy bit allocation continues until it is no longer possible to allocate bits to any of the subimages without exceeding the target rate R. The simulation defined above was run using the modified Greedy algorithm for bit allocation. Table 4.5 shows the bit allocations made with the modified Greedy algorithm, where the symbols used represent the same quantities as in Table 4.4. For each allocation, the overall measured rate $\hat{R}$ matches the target rate R, and the maximum number of bits have been

Table 4.5: Modified Greedy Bit Allocations for Boat (64×64)

Allocation			Rate		MSE
r_0	r_1	r_2	$\hat{R}$	R	D
0	4	0	1.000	1.000	228.617
1	4	0	2.000	2.000	126.638
2	4	0	3.000	3.000	47.995
3	4	0	4.000	4.000	14.830
4	4	0	5.000	5.000	4.351
5	4	0	6.000	6.000	1.279
6	4	0	7.000	7.000	0.342
7	4	0	8.000	8.000	0.093

allocated. Note the reduction in overall MSE distortion D, from Table 4.4, resulting from the use of the modified Greedy algorithm for bit allocation. The bit allocations in Table 4.5 are plotted against the full set of possible rate-distortion points for comparison in Figure 4.8. Clearly, the modified Greedy algorithm shows an improvement only in the sense that it reduces the overall distortion D, while remaining below the target rate R. In fact, from Figure 4.8, it can be seen that the modified Greedy algorithm is suboptimal at some rates, where it does not give the allocation resulting in the lowest distortion D for a given rate $\hat{R}$.

4.2.3 Simulations

To analyze the performance of lossy pyramid coding, a five level pyramid was generated from "Lenna" (see Figure B.1), in this case using decimation and interpolation filters defined by $(\sigma_d, \sigma_i) = (0.3, 0.3)$ (see Section 4.2.1). The results of this simulation for the case of other pyramid filter pairs will also be discussed later. For a given target rate R, the modified Greedy algorithm was then used to allocate bits to the subimages of the pyramid, resulting in an overall rate $\hat{R}$. After designing optimal uniform quantizers (see Section 3.6.2) based on the bit allocation, the subimages were quantized. Quantization error feedback was used during this procedure, so that the quantization error from a given level of the pyramid was fed back to the next lower level prior to the design of its quantizer, and so forth. The entropy(A.8) of each pyramid subimage i was used in place of r_i (4.1) to calculate the pyramid entropy E,

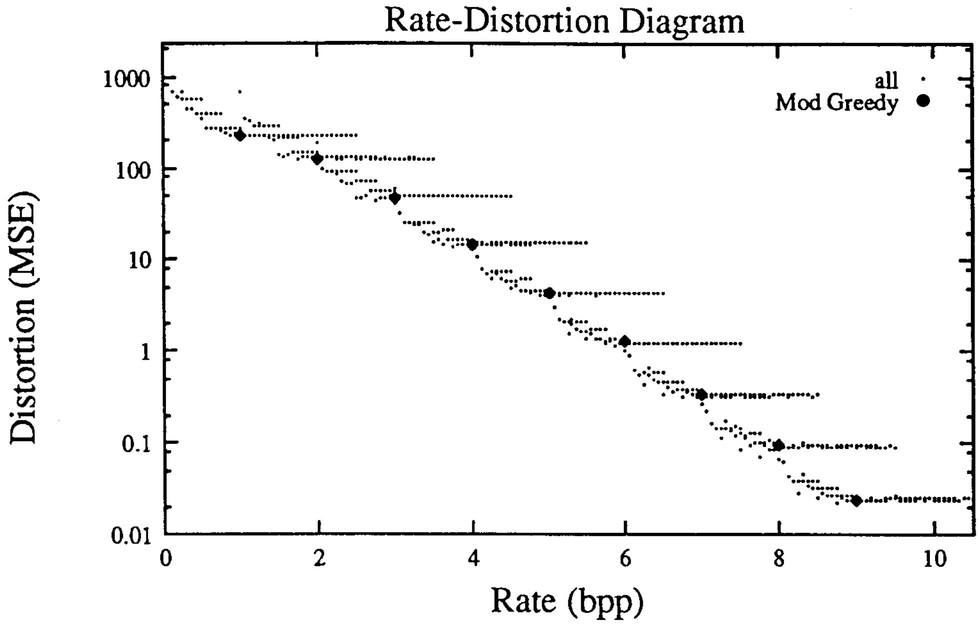


Figure 4.8: Modified Greedy Rate-Distortion Performance for Boat 64 × 64

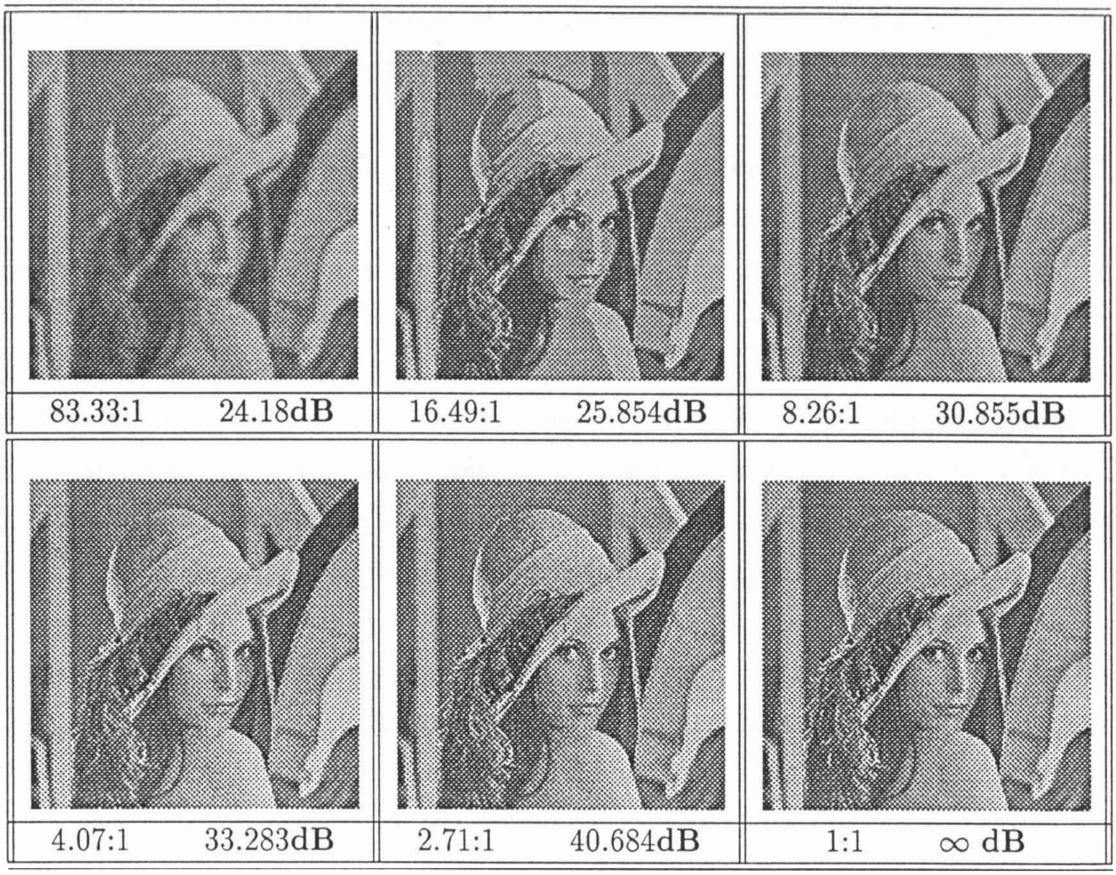
representing a lower bound on the overall rate of the pyramid after optimal codeword assignment. This lead to calculation of the optimal compression ratio C (A.14), based on the 8 bpp original image. Finally, images were reconstructed from the quantized pyramids and the distortions D measured as the PSNR (A.13), when compared to the original image. The results of this simulation are summarized in Table 4.6. For

Table 4.6: Lossy Pyramid Coding for Lenna (512 × 512)

Allocation					Rate		E	C	D
r_0	r_1	r_2	r_3	r_4	$\bar{R}$	R			
0	0	1	2	1	0.098	0.100	0.096	83.33	24.183
0	2	0	0	0	0.500	0.500	0.485	16.49	25.854
0	4	0	0	0	1.000	1.000	0.968	8.26	30.855
1	4	0	0	0	2.000	2.000	1.968	4.07	33.283
3	4	0	0	0	4.000	4.000	2.949	2.71	40.684

each allocation, Figure 4.9 shows the images reconstructed from the corresponding quantized pyramids. The original "Lenna" image is also shown for comparison.

High spatial frequency detail is reproduced relatively well in the images reconstructed from the quantized pyramids, even at lower rates. However, in regions of



83.33:1 24.18dB

16.49:1 25.854dB

8.26:1 30.855dB

4.07:1 33.283dB

2.71:1 40.684dB

1:1 ∞ dB

Figure 4.9: Lossy Images Created from Lenna 512×512

smoothly varying surfaces, mild contouring is visible. It is interesting to note the differences in the subjective quality of the first and second allocations, resulting in overall rates of 0.098 and 0.500 bpp respectively. While the former allocation results in a smooth varying perceptually pleasing image, the latter allocation gives an image that contains annoying contouring effects. This is due to the fact that the effects of quantization in the former allocation are more "smoothed" as a result of the larger number of interpolations, subsequent to quantization, done in the image reconstruction. Consequently, although the latter allocation results in a reconstructed image with a higher PSNR, it has a lower subjective quality. Although the mild contouring effects present in the reconstructed images lead to a degradation in perceptual quality, their effect is not as severe as, for example, the blocking distortion that is characteristic of the DCT at low rates. Clearly, further work needs to be done to improve the correlation between subjective and objective measures of image quality.

This however, is beyond the scope of the present research. At higher rates, corresponding to compression ratios lower than 4, distortion in the reconstructed image is nearly invisible.

When the same simulation was repeated using different pyramid filters, similar results were obtained. In fact, in most cases the allocations made by the modified Greedy algorithm were the same for a variety of different pyramid filter pairs. An exception to this was observed where interpolation filters with high σ_i were used. However, both the objective (PSNR) and subjective quality of the images reconstructed from the quantized pyramids varied considerably with the choice of pyramid filters.

In particular, for a very low choice of σ_d the reconstructed images were severely blurred, since the decimation filter was too lowpass and did not allow sufficient image information to propagate to the smaller pyramid subimages. Contouring effects were also more visible in this case. On the other hand, for a very high choice of σ_d, aliasing noise was clearly visible for some of the reconstructions, particularly in regions of high spatial frequency detail. For low overall rates, and severe quantization of the pyramid subimages, a very low choice of σ_i lead to an improvement in perceptual quality of the reconstructed images, since the severe contouring effects otherwise present were less obvious. However, a very high choice of σ_i lead to reconstructed images that contained severe noise, resulting from insufficient attenuation of the "interpolation image" distortion from the upsampling operation. These results correlated well with expectations from the analysis early in Section 4.2. It should be noted here that the MEP (see Section 4.1.1) loses much of its advantage in such a lossy pyramid coding scheme since, with quantization feedback, zeros previously guaranteed in the MEP representation may no longer be present. Consequently, lossless subsampling is no longer possible.

When the same simulation was done, except with no quantization feedback in the pyramid generation, inferior results were observed. Not only was the PSNR of the image reconstructed from the quantized pyramid lower for a given rate, but it generally contained grainy artifacts that were perceptually annoying, especially in regions of smooth transition. This observation is clearly in alignment with the analysis presented in Section 3.6.3. However, using the modified Greedy algorithm, bit allocation amongst the subimages in this case was observed to be much more uniform. Lossy pyramid coding without quantization feedback is therefore more robust to such problems as the abovementioned contouring, resulting from insufficient bit allocation to the higher pyramid levels.

A comparison of the results in Table 4.6 with results achieved using JPEG lossy coding is shown in Table 4.7. It is apparent that the performance of lossy pyramid coding is inferior to that of JPEG. Similar results were observed for other test images. However, it may be possible to further improve the rate-distortion performance of lossy pyramid coding through at least two methods in order to make it more competitive with other techniques, for example JPEG.

Table 4.7: Comparison of Lossy Pyramid Coding with JPEG : Lenna

Rate	PSNR (dB)	
(bpp)	Pyramid	JPEG
0.100	24.183	26.500
0.500	25.854	34.400
1.000	30.855	37.000
2.000	33.283	41.000
4.000	40.684	48.000

Firstly, more elaborate quantization schemes, for example vector quantization or the DCT, could be used for the quantization of the pyramid subimages. In both of these schemes, interpixel correlation may be exploited during quantization to achieve a higher coding gain and thereby reduce the distortion at a given rate.

Secondly, Farvardin and Modestino (1984) have proposed an improved method for the design of uniform quantizers, subject to an entropy constraint. In summary, the proposed technique shows that, for various standard distributions, a significant performance improvement is realized at a given rate through the use of larger quantizers, that have more quantization cells. Here, performance is measured by the variance of the quantization noise, and rate as the entropy of the quantized source. In other words, if a uniform quantizer of a given number of cells is used to quantize a source and thereby achieve a certain rate, Farvardin and Modestino (1984) have shown that a larger uniform quantizer, with a suitable load fraction, may be used to achieve the same rate with a significant reduction in the variance of the quantization noise. This was found this to be true for a variety of standard distributions, for example, the Gaussian, Laplacian, gamma and uniform distributions. However, as will be discussed later, this may not be the case for less "well behaved" distributions.

Figure 4.10 shows pixel distributions for both the pyramid base and top of a two level Laplacian pyramid generated from "Lenna" (see Figure B.1) using pyramid filters corresponding to σ_d, $\sigma_i = 0.3, 0.3$. While the base subimage has a distribution that is similar to a Laplacian standard distribution, the top subimage, a decimated version of the original image, does not have such a well behaved distribution.

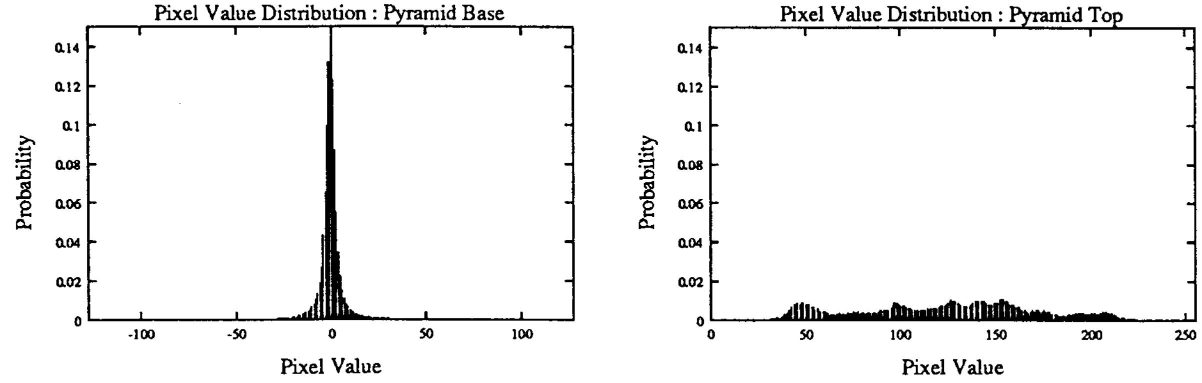


Figure 4.10: Example Pixel Value Distributions for the Subimages of a Two Level Laplacian Pyramid

The traces in Figure 4.11 show the rate-distortion performance of a 127 cell uniform quantizer. Note that any quantizer with a relatively large number of cells would suffice for the proposed technique. Here, rate is again measured as the entropy of the quantized image, while distortion is measured as the variance of the quantization noise. These traces were obtained by varying the load fraction β (2.1) of the 127 cell quantizer for the respective subimages of the two level Laplacian pyramid. On the other hand, the dots in Figure 4.11 represent the rate-distortion performance of the quantizers of rates $r_i = 1, 2, ..., 7$ bpp that were used in previous lossy coding simulations. Recall that these previous quantizers were designed with load fractions that minimized the quantization noise, irrespective of the entropy. For the base subimage, which has a approximate Laplacian pixel distribution, a significant reduction in the quantization noise may be realized through the use of the technique proposed by Farvardin and Modestino (1984). However, for the pyramid top subimage, which has an irregular pixel value distribution, the proposed technique does not lead to a significant improvement. In fact, at low rates the proposed technique leads to a decrease in performance. It should be noted here that, through the use of a larger midrise quantizer, with an even number of cells, exactly the same performance as in the case of the previous quantizers could be achieved. However, from Figure 4.11 it is apparent that little performance improvement can be realized from the new quantization method for such a pixel value distribution. The same results were achieved for both larger Laplacian pyramids and various other test images.

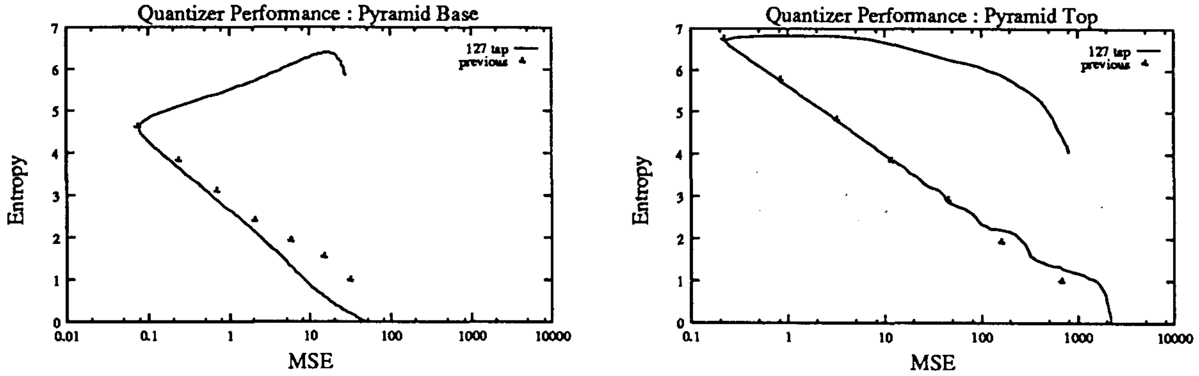


Figure 4.11: Quantizer Rate-Distortion Performance for the Subimages of a Two Level Laplacian Pyramid

Therefore, in the case of a Laplacian pyramid generated without quantization feedback, the technique proposed by Farvardin and Modestino (1984) gives a performance advantage when used on all but the top subimage of the Laplacian pyramid. In addition, through prior use of, for example, DPCM coding (see Section 2.3.1), the pyramid top subimage could be transformed into an alternative form suitable for the application of the technique of Farvardin and Modestino (1984). However, in the case of Laplacian pyramids generated with quantization feedback, this may not be the case. To understand the effect of quantization feedback on the proposed technique, consider the simple case where a two level Laplacian pyramid is generated from a test image. If no bits are allocated to the pyramid top, all pixels in the pyramid top will be quantized to the subimage mean, and a large amount of low spatial frequency information will be passed, through quantization feedback, to the pyramid base. This causes significant distortion of the well behaved approximate Laplacian pixel value distribution shown for the pyramid base in Figure 4.10. This in turn causes "bumps" in the quantizer rate-distortion performance, similar to those observed for the pyramid top in Figure 4.11, leading to an inferior performance for the proposed technique. In the case where more bits are allocated to the pyramid top, quantization feedback is less significant, and the performance of the proposed alternative technique improves.

For practical use of the technique proposed by Farvardin and Modestino (1984) in lossy pyramid coding, it is therefore desirable to measure the rate-distortion performance improvement achieved for each subimage. Clearly, if no performance improvement is realized for a given subimage through the use of the alternative technique, it should not be used in place of the previously discussed quantization techniques. Using this approach, the performance of the new quantization technique was evaluated for the 5 level Laplacian pyramid of "Lenna" (see Figure B.1), generated using σ_d, $\sigma_i = 0.3, 0.3$. The results of this simulation are shown in Table 4.8 where they are compared with previous results from Table 4.6. At some rates, a performance advantage is achieved through the use of the alternative quantization technique. It should be noted here that, ideally, this quantization technique would be integrated into the bit allocation / quantization stage of the lossy pyramid coder. However, as previously explained, this approach is complicated by the use of quantization feedback, and beyond the scope of this thesis.

Table 4.8: Modified Lossy Pyramid Coding : Lenna

Rate	PSNR (dB)	
(bpp)	Previous	New
0.100	24.183	25.830
0.500	25.854	25.854
1.000	30.855	30.855
2.000	33.283	37.024
4.000	40.684	42.031

4.3 Coding for Progressive Transmission

The conventional method employed to send an image over a channel is to encode the image line-by-line in raster fashion, from the top left of the image. However, when images must be transmitted over low bandwidth channels, this approach becomes impractical since the entire image must arrive at the receiver before it can be viewed. This may involve an unacceptable delay for slow transmission rates. In some applications, it is desirable to reconstruct full size, early approximations to the original image from subsets of information in the image representation. This allows some early interpretation of the image being sent, a feature that is particularly useful, for example, in searching image databases.

Since the pyramid is a hierarchical representation of image information, it is well suited to progressive transmission. The sensitivity of the HVS is known approximately to be a decreasing function of the spatial frequency of the information being viewed. The perceptual importance of information in the pyramid subimages therefore decreases with increasing size. Figure 4.12 illustrates progressive transmission in pyramid coding. After generation of the pyramid at the coder, the subimages are sent in order of increasing size. Upon receipt of a subimage, the decoder reconstructs the pyramid up to that level. The reconstruction may then be interpolated to full size for an early lowpass approximation to the original image. The first approximation, created by interpolation of the smallest pyramid subimage to full size, is a lowpass version of the original image. With the subsequent arrival of the larger subimages, higher spatial frequency information is added to the initial approximation until finally, after arrival of the largest subimage, the original image can be reconstructed exactly. In such a transmission strategy, it is desirable to achieve the highest possible image quality as early as possible in the transmission. Effective realization of such a strategy demands an accurate knowledge of the characteristics of the image being coded and appropriate choice of filters in both the pyramid codec and interpolation process (used

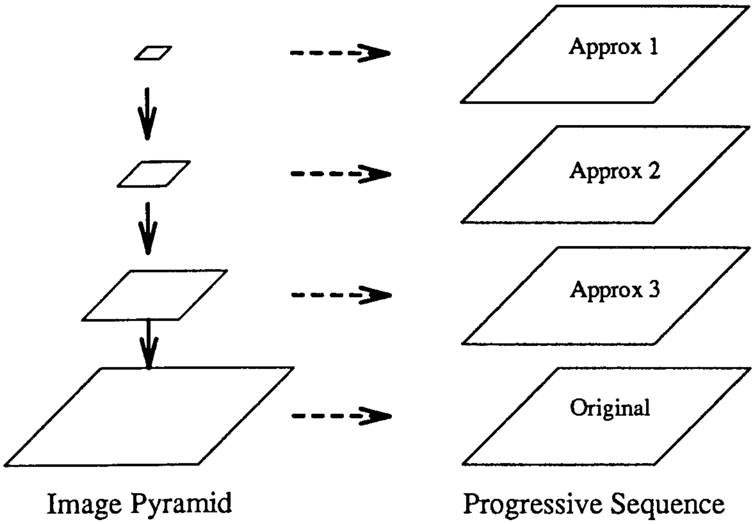


Figure 4.12: Pyramid Coding in Progressive Transmission

during progressive transmission to reconstruct the full size early approximations).

Pyramid coding for the progressive transmission of natural images will now be discussed in more detail to illustrate some of the considerations in the design of an appropriate codec. In this example, three objectives are set for the pyramid codec. Firstly, the rate-distortion performance of the progressive transmission codec should be optimized, where rate refers to the cumulative rate of the pyramid subimages during progressive transmission, and distortion refers to the quality of the full size early approximations when compared to the original image. Secondly, the final reconstruction of the image at the decoder should be lossless. Thirdly, the total rate of the pyramid should be minimized, in turn minimizing the total progressive transmission time. These objectives point to the use of MEP coding (see Section 4.1.1).

To analyze progressive transmission of the MEP, consider a two level pyramid. The top of the MEP, a subsampled version of the original image, is transmitted first. To construct an intermediate approximation, it is necessary to interpolate this subimage to full size. The interpolation filter used has an important effect on the quality of the approximation (Goldberg and Wang 1991). To find a suitable interpolation filter, it is useful to examine the subsampling, upsampling and interpolation processes in one dimension. In this case, the combined operations of subsampling and upsampling are equivalent to multiplying the time domain signal by the alternating sequence

$$x[n] = \frac{1}{2}(1 + \cos(\pi n)), \tag{4.15}$$

which is equivalent to multiplying in turn by

$$x_1[n] = \frac{1}{2} \qquad \text{and} \qquad x_2[n] = \frac{1}{2}\cos(\pi n), \tag{4.16}$$

and adding the results. The same operation may be carried out in the frequency domain by first convolving the spectrum of the original sequence with impulses at the digital frequencies $w = 0$ and $w = \pi$, and then adding these component spectra to get the resulting spectrum. This is illustrated in Figure 4.13. Clearly, the best possible approximation to the original signal contains as much of the undistorted spectrum of the original signal as possible. Ideally, the interpolation filter would pass, with a gain of two, all of the unaliased low spatial frequency information, and stop all of the aliased and "interpolation image" information. However, in practice these ideal passband and stopband filter characteristics cannot both be realized by a single filter.

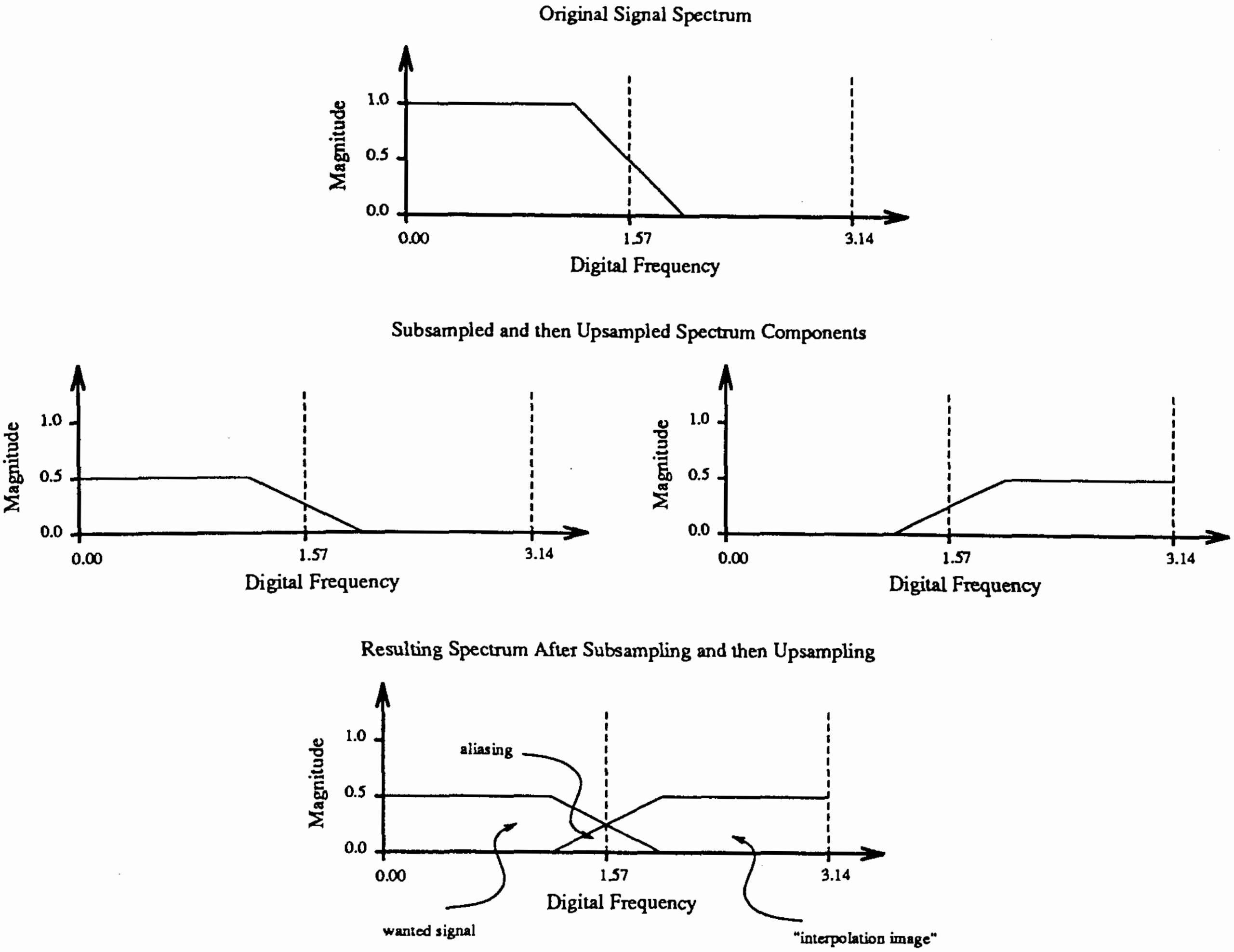


Figure 4.13: Frequency Domain Analysis of Subsampling and then Upsampling a One Dimensional Signal

A tradeoff is therefore required between the passband and stopband performances of the interpolation filter. This tradeoff is conveniently made with σ_i in the class of filters specified by (3.16). The extent of the aliasing, shown in Figure 4.13, depends on the frequency content of the original signal. Signals with mainly lowpass information will show aliasing at higher frequencies than signals with significant highpass information. Therefore, an interpolation filter with a higher σ_i will be more suitable for a signal with mainly low spatial frequency information and vice versa. A similar approach can be applied in the case of two dimensional signals, where the interpolation filter is separable and has a rectangular region of support. This concept is easily extended to pyramids with more than two levels. It should be emphasized that the choice of these interpolation filters has no effect on the final reconstructed image, which is lossless. The purpose of these filters is merely to improve the quality of the intermediate approximations.

4.3.1 Simulations

After generation of the MEP, the subimages can be sent in order of increasing size to the receiver. MEP decoding can then be done at the receiver up to the level of the last received subimage. This intermediate image can in turn be interpolated to give a full size early approximation of the original image (see Figure 4.12). As previously discussed, the type of interpolation filter used has a significant influence on the quality of the reconstructed approximation. To investigate this effect, filters from the class specified by (3.16) were used to interpolate early approximations of the test images "Lenna" and "Mandrill" in Figure B.1 during a simulated progressive transmission. The quality of an approximation was measured as its PSNR (A.13) when compared to the original test image.

Figure 4.14 shows the relationship between the PSNR and σ_i for three different approximation levels and 2 images ("Lenna" and "Mandrill"). The PSNR drops for low values of σ_i due to the fact that the interpolation filters are too lowpass and severely attenuate useful low spatial frequencies. At high values of σ_i, a drop in PSNR is also generally apparent because the filter does not adequately attenuate unwanted frequencies resulting from aliasing and interpolation. The best interpolation filters, in terms of maximizing the PSNR of the intermediate approximations, are indicated by arrows in Figure 4.14. As expected, the "optimal" σ_i is dependent on the spatial frequency content of the original. These results are reinforced by the analysis previously presented. The sequence of images corresponding to the optimal tradeoff points described above are shown in Figure 4.15. Good quality approximations to the 8 bpp test images are obtained at rates of only 0.12 bpp. It should be noted that the PSNR is not necessarily the best measure of the subjective image quality; however, in these simulations, the two measures do correlate well.

From a practical point of view, optimal interpolation filters could be pre-calculated during the image coding operation and sent to the receiver preceding the image data.

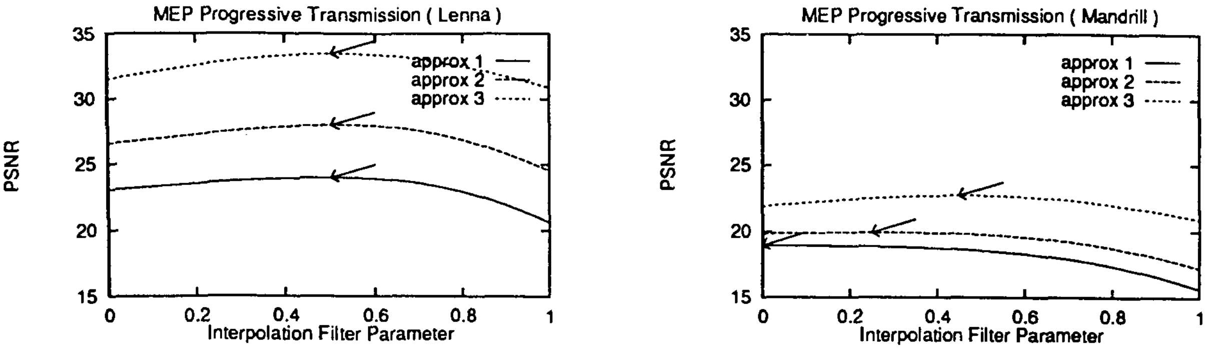


Figure 4.14: PSNR versus Interpolation Filter Type of Early Approximations of Test Images

Since the curves of PSNR versus interpolation filter type are generally smooth, numerical solution techniques could be efficiently applied to locate the optimal tradeoff points more rapidly. In the case of the MEP interpolation filter ($\sigma_i = 0.5$), the filtering operation can be carried out with only integer operations at a substantial saving in computational cost. However, other interpolation filters require the use of floating point operations and are therefore more computationally expensive to implement. This is not generally a problem though, since progressive transmission is commonly used with slow channels, allowing the receiver more time to generate the intermediate approximation images during transmission. Good progressive transmission performance has also been observed for a much wider range of test images. However, due to the fact that no anti-aliasing filtering is done in the MEP coding operation prior to image subsampling, progressive transmission performance will drop for images with significant high spatial frequency content.

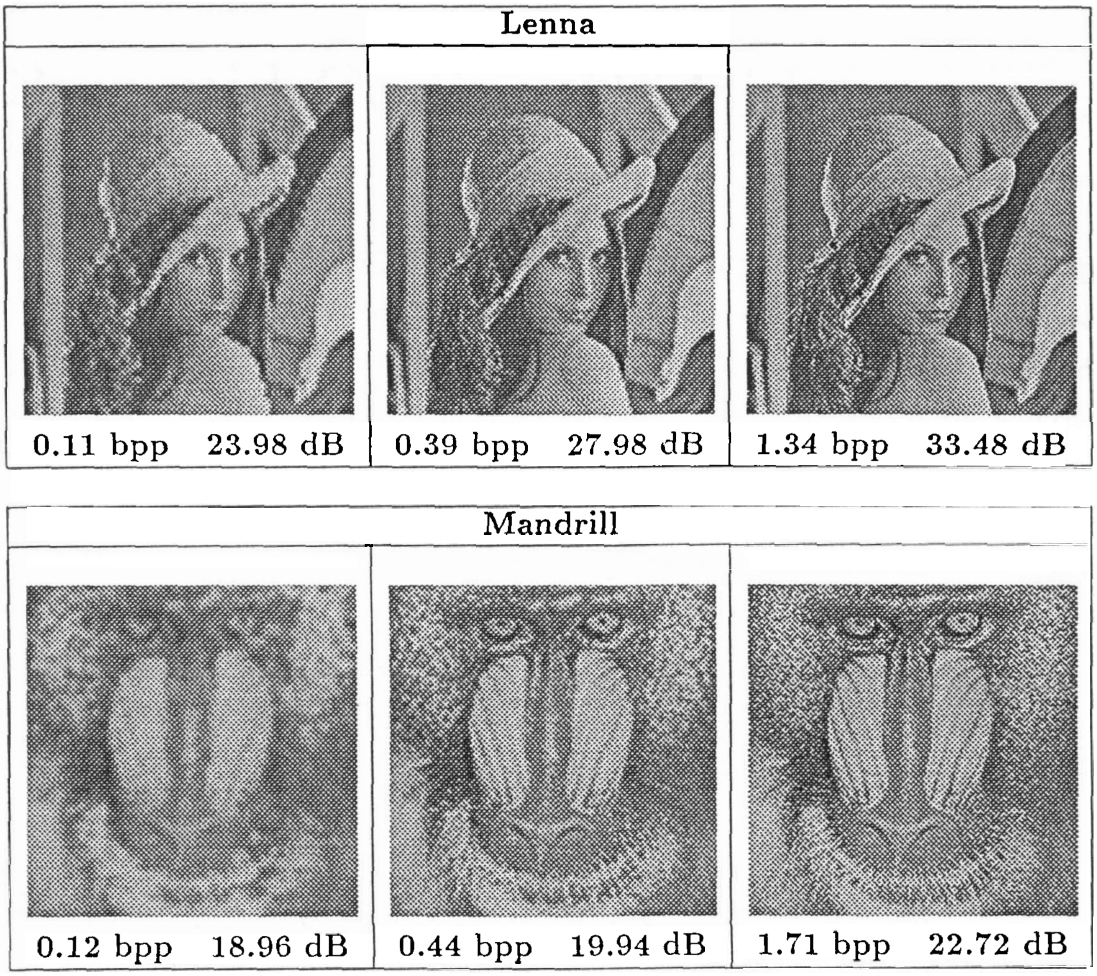


Figure 4.15: MEP Progressive Transmission Rate Distortion Performance

Chapter 5

Pyramid Coding Of Video

In this section, the use of pyramid coding in video applications will be investigated. After a brief discussion of the characteristics of video signals, a current video source coding technique, called *motion compensation* (MC), will be presented. Methods for recovering the motion information, or *motion field* (MF), required for MC will then be discussed. This will be followed by an overview of previous research and a motivation for using pyramids for more efficient MF recovery from the video sequence. A proposal for a hierarchical, pyramid based, MF recovery scheme will then be made. Finally, the proposed technique will be evaluated to determine its performance relative to both the optimal, and other suboptimal MF recovery techniques, as well as its robustness in the presence of noise.

5.1 Motion Compensation

As described in Section 1.1, a progressive video signal (see Figure 1.1) consists of a sequence of frames Ω_i displayed in rapid succession at regularly spaced time intervals. Each frame generally contains both statistical and perceptual redundancies that, as in the case of still images, may be exploited using various intraframe or spatial coding techniques. As well as this, neighboring frames in the video sequence are typically highly correlated, leading to a large amount of temporal statistical redundancy. Temporal perceptual redundancies also exist in the video sequence as a result of the temporal lowpass filtering characteristic of the HVS, and its consequent inability to see rapid interframe changes for example. MC is often used to exploit temporal redundancy in the compression of video signals. The principles of MC will now be discussed in more detail.

The objective of MC (Hsing 1987), (Sezan and Lagendijk 1993) is to minimize the temporal correlation in the frames of a video sequence (see Section 1.3). This is achieved through prediction of a given (target) frame from neighboring (source) frames in the video sequence. The most common implementation of MC is causal, and involves estimation of the target frame from the single source frame immediately preceding it. This is the form of MC that has been implemented in this research. To perform this estimation, a MF can be calculated, describing the way in which the target differs from the source, thereby allowing the target to be interpolated from the source. The interpolation error, or DFD, can then be used together with the MF to perfectly reconstruct the target frame from the source frame. For certain kinds of motion, and sufficiently accurate MF calculation, this interpolation will closely represent the target frame, leading to a low energy DFD and correspondingly high coding gain. After reconstruction, the target frame is used as the source frame for the next interpolation and so forth.

In real implementations, channel errors may be introduced into either the MF or DFD component of the MC signal. To limit the propagation of these errors, MC typically incorporates some form of signal replenishment. For example, every n^{th} frame may be sent without MC, perhaps with some form of spatial coding. Alternatively, a more even video rate can be achieved by sending a different portion of each frame spatially coded. For example, a column or row may be sent spatially coded with each MC coded frame to replenish the video signal in a sweeping fashion. Similar replenishment techniques are used in, for example, MPEG to allow random access to the video sequence.

In an efficient MC scheme, the MF may be calculated accurately at relatively low computational cost, and both the MF and DFD may be efficiently coded for a high compression. To implement an efficient MC scheme, a number of assumptions are made. Firstly, it is assumed that the same point on an object in different frames of the video sequence will have the same intensity. This assumption generally holds true to a close approximation in practice, but fails with scene lighting changes, reflections or shadows for example. Secondly, the rigid body assumption requires that an object maintain the same shape from frame to frame so that the motion of points in the same object can be closely approximated using a single motion vector. This also requires that any apparent motion in the video sequence is translational. Rotational and zoom motion for example are exceptions to this, but may be approximated as translational motion with sufficiently small block sizes, and a finer MF. MC coders also assume that neighboring frames are highly correlated, allowing a given frame to be interpolated from its predecessor. This is generally true in practice, but will not hold during scene changes in a video sequence. During such events, the entropy of both the MF and DFD increase, leading to a drop in the coding gain.

Figure 5.1 shows MC coding of a video frame. In order to exploit temporal correlation between the neighboring frames Ω_{n-1}, or source frame, and Ω_n, or target frame, the latter can be represented as an interpolation of the former. For this interpolation, a MF must be calculated that describes how Ω_n differs from Ω_{n-1}. After subdividing Ω_n into a uniform field of blocks, a search is done for each block in Ω_n to find the best match block in Ω_{n-1}, where the closeness of a match between two blocks is measured by some correlation measure. In this research, a *sum of squared error* (SSE) correlation measure (A.9) was used. In an exhaustive implementation of MF recovery, correlation measures for all possible motion vectors in a given search area are computed before the best match is determined. This technique is optimal in the sense that it always finds the best match in the defined search area. However, it generally involves high computational expense that may otherwise be avoided with the use of suboptimal guided search techniques. These techniques use information from previously calculated correlation measures to guide the search for the final motion vector, and thereby avoid the computational expense involved with the calculation of many unnecessary correlation measures. Methods for suboptimal MF recovery will be the subject of later discussion. The optimal motion vector, indicating the best match, is then stored in the MF used to interpolate each block in Ω_n. After complete recovery of the MF, consisting of the best match motion vector for each block in Ω_n, the target frame can be interpolated from the source frame Ω_{n-1}. The interpolation error, or DFD, and MF therefore completely represent Ω_n and can usually be coded at a much lower rate than the original frame.

The interframe displacement of each motion block is described by a single two dimensional vector, referred to as the motion vector. To achieve a good coding gain through MC coding, a motion vector must accurately represent the apparent motion of

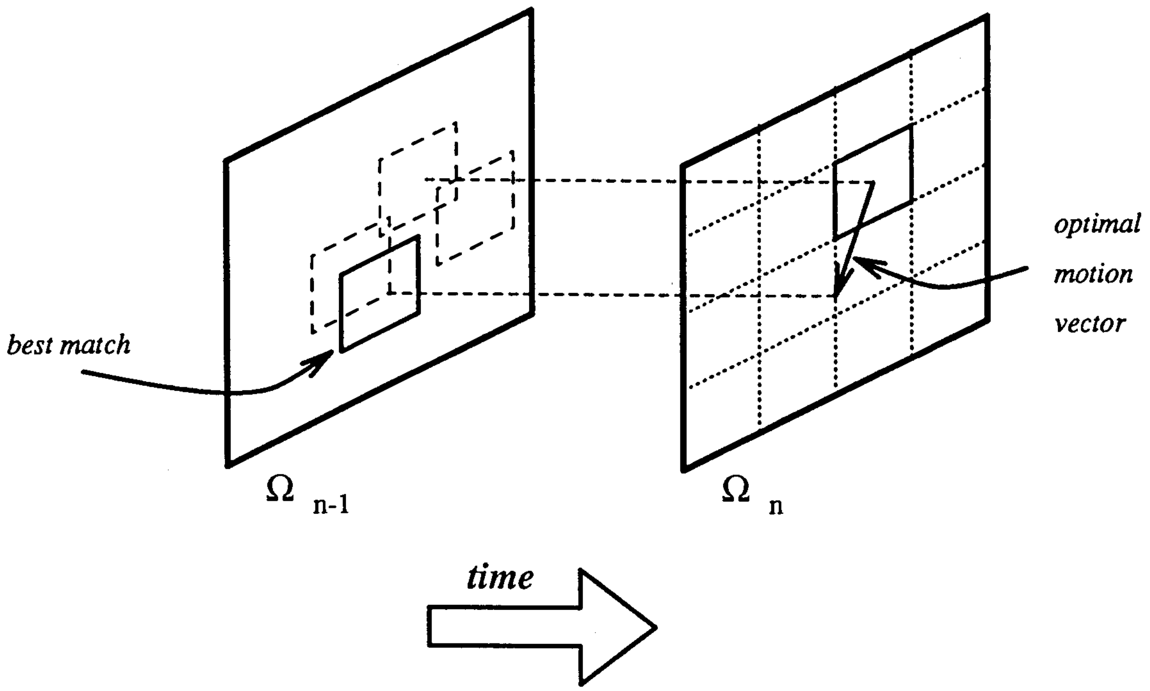


Figure 5.1: Motion Compensation

the complete interior of the motion block. Clearly, "moving" objects in practical video sequences are not always rectangular in shape, and may be smaller than the motion block size. This leads to a drop in performance, in terms of coding gain, for the MC coder. However, through appropriate choice of motion block size, and indirectly the resolution of the MF, the MC scheme can closely approximate the apparent motion in practical video sequences, and thereby achieve a good coding gain. Some of the issues involved in choice of the motion block size will now be discussed.

Later, in Section 5.3.4, it will be shown that the computational cost of implementing an MC scheme is independent of the motion block size. However, the coding gain achieved by the MC coder depends on the motion block size for a number of reasons. On the one hand, a smaller motion block size leads to a finer MF that requires more bits to encode as overhead information. Smaller blocks are also more easily matched (Bierling 1988) and therefore lead to more local minima in the SSE correlation surface, obtained while searching for an optimal motion vector for a given block. These local minima have no effect on an exhaustive MF recovery scheme, that is guaranteed to always find the global minima in the correlation surface, but they may confuse suboptimal guided-search MF recovery schemes. This causes a decrease in the quality of the MF recovered by the the guided-search techniques, increasing the entropy of both the MF and DFD. However, a finer MF can approximate the interframe motion more closely and thereby improve the interpolation of a given target frame from the source frame preceding it (see Figure 5.1). This in turn leads to a lower energy DFD. Note that in the extreme where the block size is reduced to a single pixel, the MC scheme is referred to as "pixel-based MC". MF recovery in pixel-based MC is very similar to the calculation of optical flow fields (Singh 1991), where each pixel is assigned a motion vector. On the other hand, larger blocks lead to a coarser MF that requires less overhead to represent. Although they may not represent complex motion as well as smaller block sizes, they have better behaved correlation surfaces, leading to less false matches. Clearly, there exists a tradeoff in block size that is optimal in terms of reducing the overall bit rate. However, this optimal block size is source dependent, and may vary according to the types of motion dominant in a video sequence. For practical video sequences, a block size of 8 × 8 has been found to lead to good performance. This is also the block size used in most hardware available for the implementation of MC. 8 × 8 blocks will therefore be used in this research. Akansu, Chien, and Kadur (1989) have shown that, for this block size, the losslessly encoded MF typically requires 10% – 15% of the overall MC codec bit rate.

The majority of the computation involved in the implementation of an MC coder is involved with recovery of the MF. While an exhaustive search is optimal, as previously discussed, it demands high computational expense. There exist many alternative suboptimal search techniques that make various performance versus computational cost tradeoffs. Some of these schemes will now be discussed in the form of a brief overview. This foundation will lead to the development of a new suboptimal MF recovery technique, based on pyramid coding.

5.2 Existing Schemes for Pyramid Based Motion Recovery

Pyramid coding has been used in a variety of video source coding applications both indirectly, in the efficient calculation of motion information, and directly in the coding of image frames or MC residuals. A brief overview of some of this previous research will now be given.

5.2.1 Machine Vision

Many schemes have been proposed to exploit the pyramid hierarchy in efficient algorithms for various image analysis and processing tasks, particularly in the field of computer vision for applications such as image registration, motion detection and stereo vision. For example, Glazer, Reynolds, and Anandan (1983) investigated the use of both lowpass and bandpass pyramids in hierarchical scene matching for various machine vision applications. Dengler (1986) proposed a method for efficiently calculating image "correspondence" from three level Gaussian pyramids, where the correlation measure used in the hierarchical scheme was calculated based on the sign of the Laplacian operator, when used on the pyramid subimages. Bergen and Adelson (1987) proposed a hierarchical, pyramid based, technique for recovering optical flow information from a video sequence.

Schemes based on pyramids have also been proposed for the recovery of more complex motion information, such as the motion of transparent objects, from image pairs. For example, Bergen and Burt (1990) proposed a technique, based on the Gaussian pyramid, for efficient recovery of complex optical flow information. A further refinement of this concept was made by Burt, Hingorani, and Kolczynski (1991), who presented a similar technique based on the Laplacian pyramid.

Pyramids have also been proposed for the recovery of motion information to be used for example in scene segmentation and the interpretation of three dimensional structure from image sequences. For example, Anandan (1987) analyzed methods for determining apparent motion, between image pairs, for use in computer vision applications. A hierarchical MF recovery technique, based on the Burt and Adelson (1983) Laplacian pyramid, was proposed. The Laplacian pyramid was chosen for this scheme based on the findings of Burt, Yen, and Xu (1982) that some improvement is realized through the estimation of correlation between images that have first been filtered by a Laplacian operator. However, Burt, Yen, and Xu (1982) also showed that such correlation measures were sensitive to noise in the image pair.

5.2.2 MF Recovery for Video Coding

Bierling (1988) proposed an alternative hierarchical motion field recovery technique that is related to schemes based on the Gaussian pyramid in the sense that it uses lowpass filtering followed subsampling of the pixel field to reduce computational cost. This technique estimates the motion vectors in a three stage hierarchical manner, each stage using a motion block, or "window", of a different size. The first stage uses a large window, which generally has a correlation surface with less local minima, to get a coarse estimate of the motion vector. Subsequently, the estimate is refined using smaller windows, but searching is constrained to be sufficiently localized so that the probability of a false match is minimal. To reduce the computational cost of the technique for the coarse estimates, the source and target frames are decimated by first lowpass filtering, using either a 3×3 or a 5×5 rectangular or mean filter, and then subsampling by a factor of two both horizontally and vertically.

Dufaux and Kunt (1992) have suggested a multigrid block matching motion estimation technique, where motion estimation is based on the original source and target frames, but on a set of grids with different resolutions. Motion information initially estimated based on the coarser grid is subsequently refined on the finer grids to calculate the MF in a hierarchical manner. In addition, this research suggests a quadtree decomposition of the MF into blocks of different sizes according to the complexity of the motion in a given area. This approach assigns larger blocks to areas of uniform translational motion, and smaller blocks to regions of MF discontinuity for example at object boundaries. Dufaux and Kunt (1992) showed that such an approach to MF decomposition leads to a significant reduction in the energy of the DFD, without increasing the number of motion vectors per frame to be encoded.

Chun and Ra (1992) suggested a hierarchical MF recovery technique based on mean pyramids. Their technique is centered on successive refinement of the block matching criterion. This approach differs from those centered on guided-search motion vector refinement, for example (Bierling 1988), in that it first calculates a subset of possible best match motion vectors using a full search, but with an approximated matching criterion. Subsequently, the subset is refined down to a single motion vector using a progressively more accurate matching criterion.

5.2.3 Pyramids Applied in Video Coding

At least three different schemes have been proposed for the integration of pyramid coding into the video codec. Stiller and Lappe (1991) presented a technique for lossy pyramid coding of the DFD for each MC coded frame. Uz, Vetterli, and LeGall (1991) proposed a scheme for integrating pyramid coding into video compression, where both Gaussian and Laplacian pyramids were used for hierarchical motion estimation and the spatial coding of frames respectively. More recently, Gandhi, Wang, Panchanathan, and Goldberg (1993) proposed a Laplacian pyramid based scheme for the source coding of video, both in the spatial and temporal dimensions.

5.2.4 Summary

From this foundation of research into the application of pyramid coding for both the recovery of motion information, and spatial or temporal source coding, it is clear that pyramids have desirable characteristics for video source coding. While the previous research proposes various methods for exploiting the hierarchy and spatial frequency separation of the pyramid, little attention is paid to the choice of filters to be used in pyramid generation. These filters have previously been shown to have a significant effect on the characteristics of the pyramid. Furthermore, although schemes based on both the Gaussian and Laplacian pyramid have been proposed, a thorough comparison of the relative performance of these two types of pyramid has not, to the best knowledge of the author, been done. These issues will be discussed in more detail in the next section, followed by a motivation for further research into the use of pyramid coding for efficient MF recovery.

5.3 Hierarchical MF Recovery Using Pyramids

5.3.1 Motivation

Two important advantages of the pyramid are that it is a hierarchical representation of image information, and that it can be tailored to suit a particular objective through appropriate choice of generation filters. Interframe MF recovery in MC coding of video is an application in which both of these advantages can be exploited to achieve near optimal performance at a small fraction of the computational cost required by the exhaustive search.

As will be shown in Section 5.3.4, the computational cost of the exhaustive MF recovery scheme outlined in Section 5.1 increases exponentially with the maximum interframe block displacement. It should be noted here that the actual maximum interframe displacements that objects are likely to move in a video sequence are bounded by constraints in preserving continuity of apparent motion and will probably remain constant with future video system implementations. However, the resolution in future video systems is likely to increase. Therefore the maximum interframe displacement that an object is likely to undergo, in terms of pixels, will probably increase. Future MF recovery schemes will therefore have to deal with larger displacements, and computational efficiency will become an even more serious issue. To present, the significance of this high computational cost has been reduced by the fact that most coding applications are currently broadcast in nature and highly asymmetrical in computational cost. However, with future trends of more direct user-to-user communications, this asymmetry will be intolerable in most applications involving cheap user hardware with low computational power. This is an important motivation for the use of hierarchical MF recovery schemes in future video systems.

Where the exhaustive search is impractical due to its high computational cost, other suboptimal techniques exist for the recovery of the MF. Instead of taking the "brute force" approach of computing the correlation measure for all possible motion vectors in the defined search area, these suboptimal techniques use information about previously computed correlations to guide the search for the final motion vector. In the case of well behaved correlation surfaces that have a clearly defined global and no local extrema, these suboptimal techniques perform as well as the optimal technique. However, in practice, correlation surfaces are generally not well behaved and have local minima that may confuse suboptimal schemes and lead to an increase in the entropy of both the MF and DFD, resulting in less efficient MC coding. For example, Figure 5.2 shows an SSE correlation surface, obtained for an 8×8 block size in an exhaustive search for a motion vector during MC coding of two frames of the "Ping Pong" sequence (see Figure B.2). Clearly, to achieve the best coding gain, it is desirable to minimize the number of false matches in the MF recovered by a suboptimal technique. Here, a false match is defined to be a motion vector that

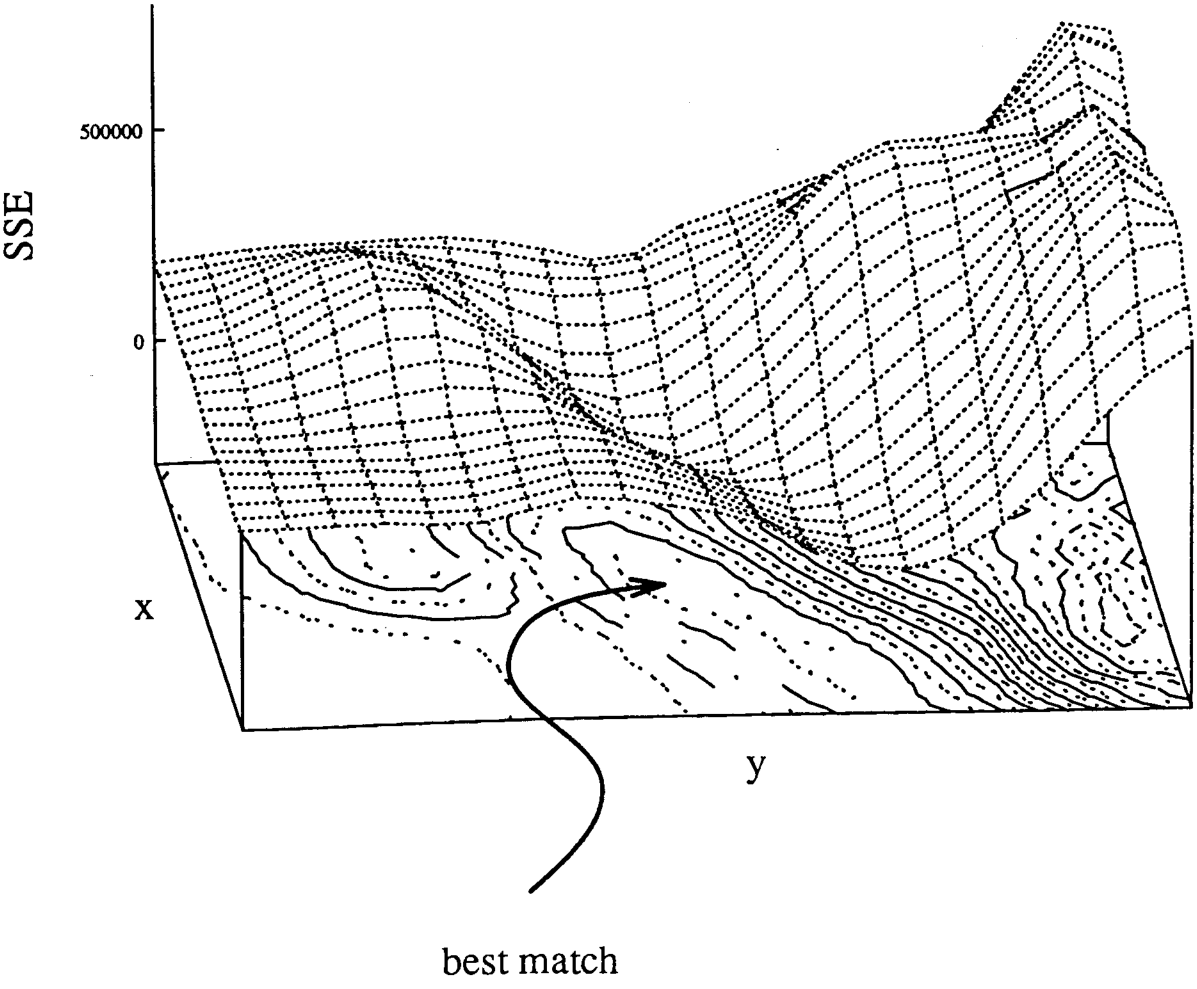


Figure 5.2: Example SSE Correlation Surface from "Ping Pong" Video Sequence

differs from the motion vector recovered by the optimal exhaustive technique.

5.3.2 An Illustration of Hierarchical MF Recovery

To understand the principle behind hierarchical MF recovery, it is useful to consider image information in the frequency domain. Figure 5.3 shows a one dimensional continuous sine wave. Clearly, for such a signal, the first false match occurs at a distance of one wavelength λ from the true match. While it is desirable to maximize the distance to the first false match, it is also desirable to maximize the accuracy of the resolved true match. For low frequency signals, λ is large, and the first false match occurs relatively far from the true match. However, for such a signal, the position of the true match cannot be resolved accurately. Note that in the extreme case of a zero frequency signal, no match can be resolved at all. On the other hand, a high frequency signal can resolve the position of the true match to a higher accuracy, but has first false matches much closer to the true match. To avoid this dilemma of conflicting objectives, it is possible to first compute a rough estimate of the true match based on the low frequency information in a signal, and subsequently refine the match based on the higher frequency information. This analysis can easily be extended to two dimensional signals, and although idealistic in the assumption of pure sinusoidal content, it illustrates the principle behind hierarchical techniques for MF recovery.

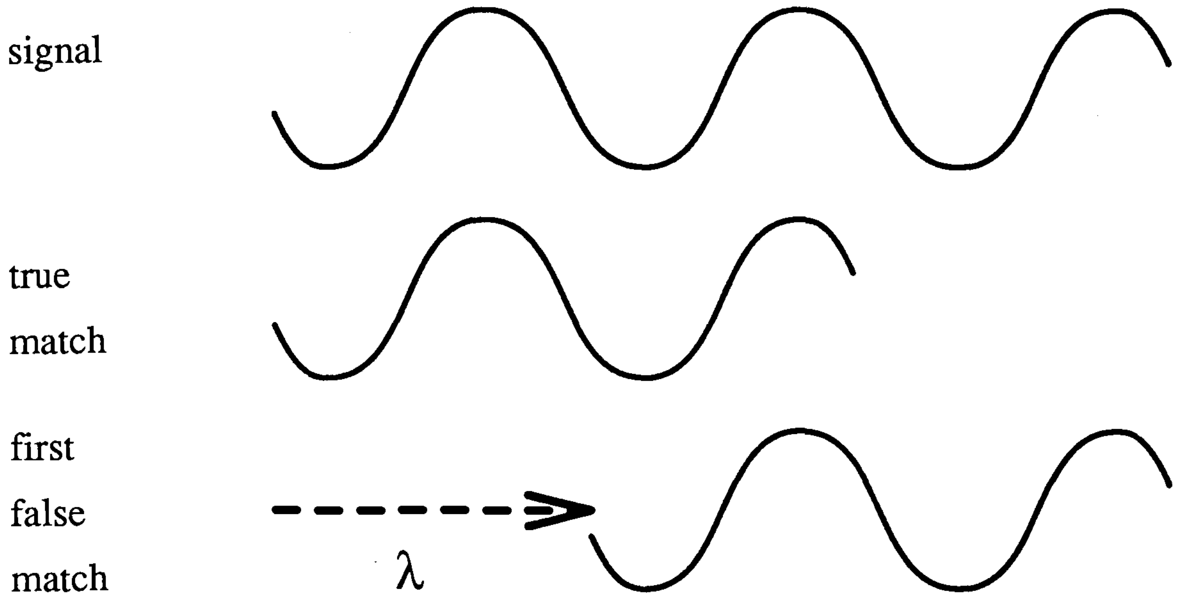


Figure 5.3: True and First False Match for a One Dimensional Sine Wave

5.3.3 Summary

In the next section, a hierarchical MF recovery scheme based on pyramid coding will be proposed for implementation in a MC coder. This will be followed by an evaluation of the performance of the proposed technique, when compared to both optimal exhaustive MF recovery and another similar suboptimal MF recovery technique, recently proposed by Zaccarin and Liu (1992). Here, performance will be evaluated in terms of both lossless MC coding gain, and computational cost. The robustness of the proposed hierarchical technique to noise will also be evaluated. Finally, the overall performance of the proposed scheme, in an example of a practical codec, will be evaluated.

5.3.4 Hierarchical MF Recovery Using Pyramids

MC coding has proven to be a valuable tool in the compression of video signals. However, the success of MC depends critically on accurate MF calculation. The optimal MF recovery scheme, in terms of finding the motion vector corresponding to the best match for each block, is the exhaustive search method. However, this is also the most computationally expensive technique and it is impractical for large interframe block displacements or image frame dimensions (see Section 5.3.4). In practice, suboptimal MF recovery schemes are often implemented, for example the three-step search (Koga, Iinuma, Hirano, and Iijima 1981) or the decimated search (Zaccarin and Liu 1992). In this section, an alternative hierarchical MF recovery scheme, based on pyramid coding, will be presented. In the development of this technique, such issues as the pyramid type (Gaussian or Laplacian), number of pyramid levels, pyramid filters and computational cost will be considered in optimizing its performance. It will be shown that the computational cost of this scheme is far lower than that of both the exhaustive search method, as well as several existing "fast" suboptimal techniques, making it more practical for real time applications.

Gaussian pyramids (see Section 3.2) have been more widely applied than Laplacian pyramids (see Section 3.3) in hierarchical techniques for the calculation of optical flow (see Section 5.2). They have the advantage that they are cheaper, in a computational sense, to generate (see Section 3.2) and do not contain the extra distortion introduced by the interpolation filtering operation, in the form of both the unwanted "interpolation image" and computational noise. However, arguments have been presented suggesting that the Laplacian pyramid may be better suited to a hierarchical search of this form. For example, Anandan (1987) suggested that Laplacian pyramids are more suitable for hierarchical motion estimation because they offer greater separation of the spatial frequencies in the images being processed. As well as this, the Laplacian pyramid was so named, as described in Section 3.3.1, due to similarity of the Laplacian pyramid generation iteration and the Laplacian operator, widely used for edge detection in computer vision. Edges and other high spatial frequency information, that are enhanced by the Laplacian operator, represent image features that lead to the unambiguous recovery of best match motion vectors. However, the Laplacian operator attenuates low spatial frequencies that help to define the best match at larger displacements. Therefore, it is even more important in the case of the Laplacian pyramid to constrain the maximum block displacement to small values during the intermediate stages of the hierarchical search.

In Figure 5.4, the proposed hierarchical MF recovery scheme for a three level pyramid is shown diagrammatically. Either Gaussian or Laplacian pyramids are initially generated from each of the two neighboring frames Ω_{n-1} and Ω_n as in Figures 3.2 and 3.3 respectively. Δ_i^j denotes the MF for the i^{th} frame, at the j^{th} level of the pyramid. R and $\uparrow 2$ denote the MF refinement and MF interpolation operations respectively. In order to evaluate the optimal performance of the proposed hierarchical scheme, no quantization of the subimages l_i^j is done prior to MF recovery. The subimages of the pyramids therefore generally consist of pixels with a wide range of floating point values. Note, however, that the effect on the proposed scheme of quantization noise in the pyramid will be evaluated in subsequent simulations.

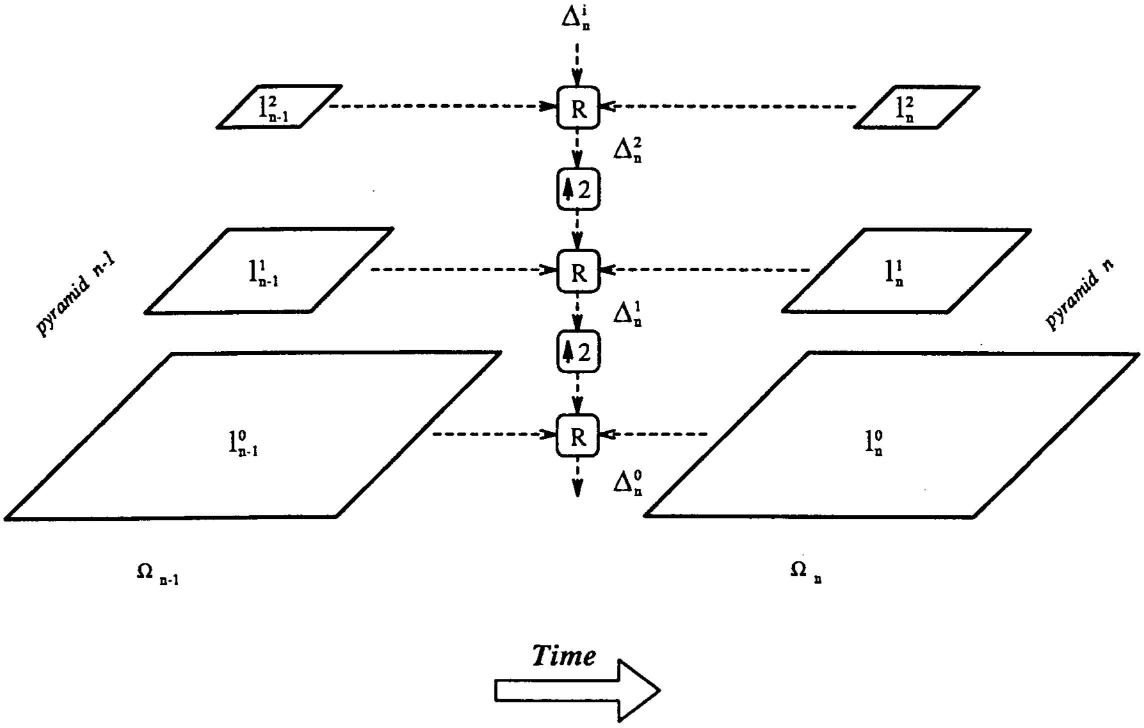


Figure 5.4: Proposed Scheme for Hierarchical Motion Field Recovery

MF refinement consists of an exhaustive search within a small area centered on the initial motion vector. The motion vectors corresponding to the new best matches are stored in the refined MF.

MF interpolation, shown in Figure 5.5 consists of upsampling a MF by a factor of two, interpolating "missing" vectors using a nearest neighbor approach, and finally scaling all the vectors by a factor of two. It is possible that this interpolation could be done differently with some improvement in performance using upsampling and lowpass filtering; however, the proposed MF interpolation has the advantage that it is computationally inexpensive, a central issue in the proposed scheme. In fact, motion vectors consist of integer components, and since the scaling in the interpolation process is by a factor of two, bit shift operations may be used instead of more costly floating point multiplies. The maximum overall displacement D_N that can be computed by the hierarchical MF recovery scheme, using N level pyramids, is given by

$$D_N = (2^N - 1)d_h, \tag{5.1}$$

where d_h is the maximum block displacement used in the intermediate MF refinements in the proposed scheme. After generation of the pyramids, a MF is computed

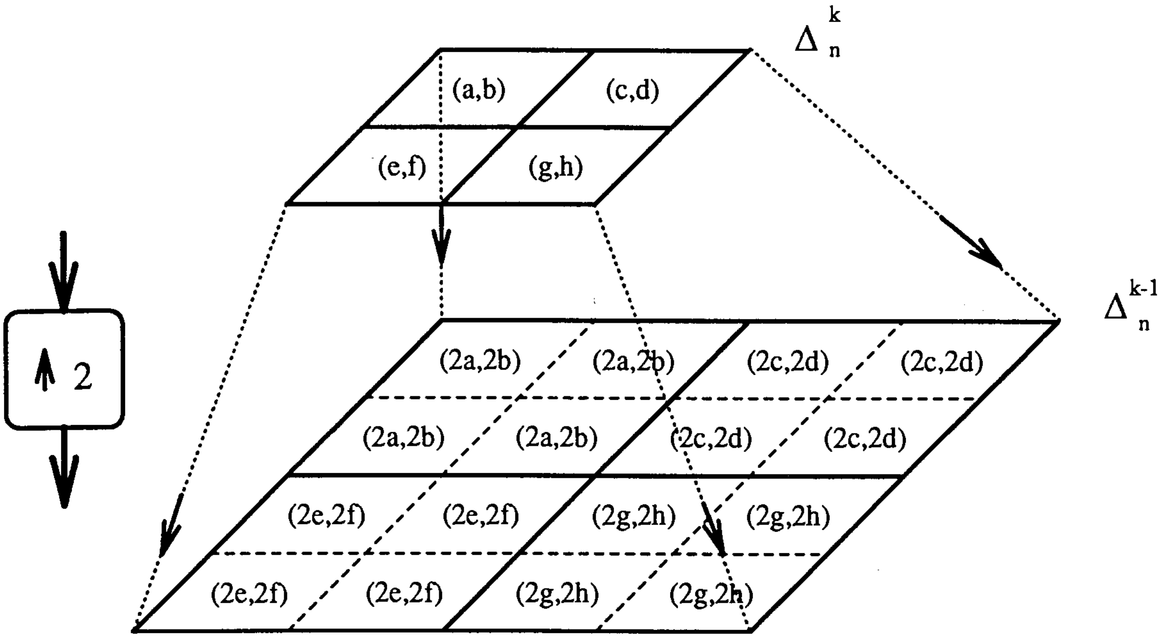


Figure 5.5: Motion Field Interpolation

hierarchically using the subimages of the pyramids. In the initial MF, Δ^i_n, the vectors are set to either zero or some estimate. Δ_n^i is fed into the first MF refinement stage where the smallest subimages l_i^2 are used to get a coarse initial estimate Δ_n^2 of the MF. Δ_n^2 is then interpolated and further refined using the subimages l_i^1 of the pyramid to produce Δ_n^1. Finally, Δ_n^1 is interpolated and refined using the largest subimages l_i^0 to give the final estimated MF Δ_n^0. Δ_n^0 and Ω_{n-1} can be used to interpolate Ω_n. The difference between Ω_n and the interpolation, or DFD, can then be encoded along with Δ_n^0 as sufficient information to completely reconstruct Ω_n from Ω_{n-1}.

It is desirable to limit the maximum displacement, searched at each level of the hierarchy since, due to the lowpass filtering, as well as the aliasing and computational noise introduced in the decimation process, the reliability of coarse motion vector estimates decreases with the size of the pyramid subimages on which they are based. Therefore, the probability of bad[1] matches at higher levels of the pyramid is greater, particularly at larger displacements. A bad match that occurs early in the hierarchical search is undesirable since it "steers" the estimator away from the desired optimal motion vector. To minimize the probability of bad matches, the maximum intermediate block displacements should be limited to define only a small search area in which the best match is likely to be found.

In the next section, it will be shown that the filters used in the pyramid generation have a significant effect on the success of the proposed technique, when based on either the Gaussian or Laplacian pyramids. Pyramid filters well suited to hierarchical MF recovery will then be presented.

Pyramid Filters

It is desirable to find the pyramid filters (see Section 3.5.1) that are best suited to the hierarchical MF recovery scheme proposed in Section 5.3.4, for the case of both the Gaussian and Laplacian pyramids. The optimal exhaustive MF recovery scheme will be used to provide a bound on the best performance achievable by the proposed scheme.

For various pyramid filters from the classes defined by equations 3.10 and 3.16, a simulation was done on the standard video sequence "Miss America" (see Figure B.2) to evaluate the performance of the proposed scheme. For each pair of neighboring

[1] A bad match refers to a match that leads the hierarchical estimator away from the optimal match that would result from the use of the exhaustive search technique.

frames in the sequence, the MF was recovered with the proposed hierarchical scheme, using three level pyramids, a maximum intermediate block displacement d_h of two pixels and a zeroed initial MF Δ_n^i (see Figure 5.4). The optimal MF was then recovered with the exhaustive scheme, using a maximum interframe block displacement D_N (5.1). From each MF, the corresponding DFD for the target frame was calculated. The combined entropy E_{total} of the MF and corresponding DFD was then calculated for both cases as

$$E_{total} = E_{dfd} + \frac{N_{mv}}{N_p} E_{mf}, \tag{5.2}$$

where N_{mv} and N_p are the number of motion vectors in the MF and the number of pixels in each frame of the video sequence respectively. E_{dfd} and E_{mf} are the entropies of the DFD and MF for the target frame, calculated as straightforward extensions of equation A.8. Finally, the difference in E_{total} for the hierarchical and exhaustive MF recovery schemes was calculated.

This difference was averaged over the "Miss America" video sequence (see Figure B.2) and is plotted in the case of the Gaussian and Laplacian pyramids in Figures 5.6 and 5.7 respectively, as a function of the pyramid filters. Similar results were observed for the same simulation on the "Ping Pong" video sequence. The entropy difference in these figures indicates the penalty in performance, measured as the average rate in bits per pixel, for using the hierarchical MF recovery schemes, when compared to the corresponding optimal exhaustive MF recovery scheme. Note that the pyramid filters influence the success of the proposed scheme. A drop in performance can be seen for large σ_d in both types of pyramid, indicating that the proposed technique is sensitive to aliasing noise introduced by the subsampling operation in the decimation process. Note that, for this reason, the MEP (see Section 4.1.1) is unsuitable for use in such a pyramid based MF recovery scheme. A similar drop can be seen in the case of the Laplacian pyramid for large σ_i, indicating that the technique is also sensitive to "interpolation image" noise resulting from the upsampling operation in the interpolation process. Optimal performance of the hierarchical MF recovery scheme corresponded to a decimation filter choice of $\sigma_d = 0.0$ for both types of pyramid, and an interpolation filter choice of $\sigma_i = 0.0$ for the Laplacian pyramid. These are the pyramid filters that have the best stopband performance (see Section 3.5.1), and therefore minimize the two types of noise in the pyramids.

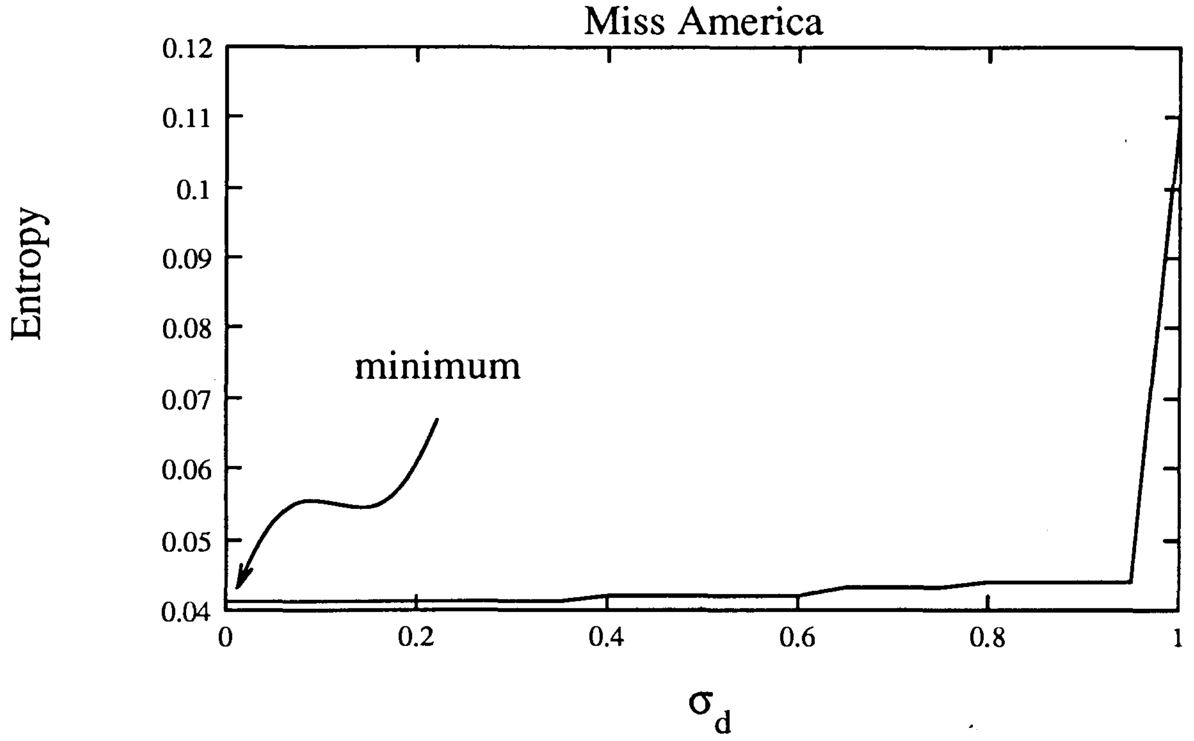


Figure 5.6: Hierarchical MF Recovery from 3 Level Gaussian Pyramids ($d_h = 2$)

The optimal average combined entropy, based on exhaustive MF recovery, was computed to be 3.337. For the Gaussian pyramid, the best performance of the hierarchical scheme, using the decimation filter discussed above, was computed to be 3.379, representing a performance penalty of only 1.25%. On the other hand, for the Laplacian pyramid, the best performance of the hierarchical scheme, using the pyramid filter discussed above, was computed to be 3.474, this represents a performance penalty of approximately 4.1%. Similar results were observed for other video test sequences. In the next section, it will be shown that this near optimal performance is attained at a small fraction of the computational expense required by the exhaustive MF recovery scheme.

Calculation of the SSE correlation measure (A.9) involves summation of the squared error over the block area. This has a lowpass filtering effect that becomes more significant with larger block sizes, and tends to desensitize the effect of the pyramid filters on the overall performance. For example, when the simulation described above was run with a smaller block size of 4×4 pixels, the effect of the pyramid filters on

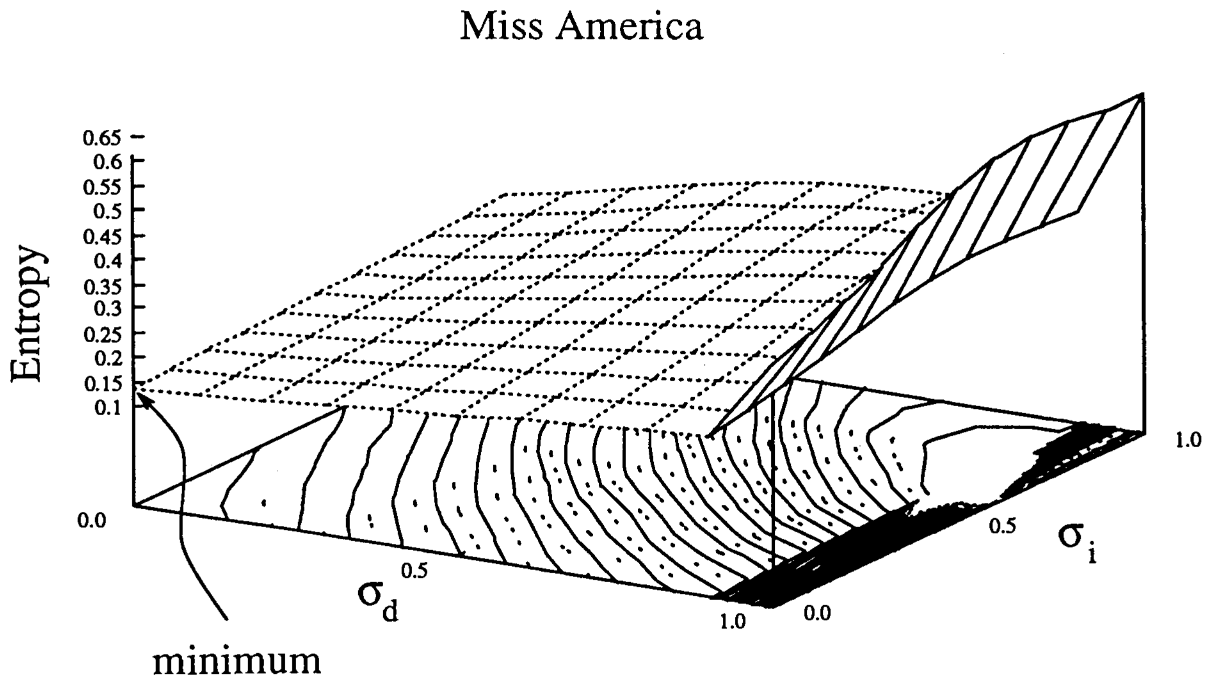


Figure 5.7: Hierarchical MF Recovery from 3 Level Laplacian Pyramids ($d_h = 2$)

the performance of the hierarchical MF recovery was found to be more significant. However, for both choices of block size, the optimal filters were found to be the same. Furthermore, the best performance achieved using the 8×8 block size was found to be superior to that achieved when using the 4×4 block size, as expected from the discussion of the effect of block size presented in Section 5.1.

Hierarchical MF recovery methods based on mean pyramids have been proposed by, for example, Bierling (1988). Therefore, for the sake of comparison, it is of interest to compute the performance measure used for the previous simulation for the case where the decimation filter is the rectangular 5×5 mean filter used by Bierling (1988). Using the same simulation parameters as for the Gaussian pyramid discussed above, the simulation was again run on the "Miss America" sequence to determine the performance of the rectangular filter. The results of this simulation revealed that, for this video sequence, the rectangular filter had the equivalent performance, in terms of combined entropy (5.2), as the filter corresponding to $\sigma_d = 0.85$. From Figure 5.6 it is clear that this is not the best filter to use, in the proposed scheme, in terms of the chosen performance measure. Similar results were observed for the "Ping Pong" sequence. However, the rectangular mean filter does have the advantage that it may be implemented at a lower computational cost using only additions. Note that the above simulation does not attempt to make a direct comparison of the technique proposed above with that proposed by (Bierling 1988). It does, however, indicate that the mean filter is clearly not the best choice, in terms of the chosen performance measure, for use in the scheme proposed above. It is likely that this is a result of the bad stopband performance of the mean filter, when compared to that of the ideal decimation filter (see Section 3.4).

A relatively large drop in performance can be seen, in the case of both pyramids, for impulse decimation filters (4.2) corresponding to $\sigma_d = 1.0$. Note that the decimation process in this case consists only of subsampling. Many existing suboptimal search techniques, for example see Section 5.3.5, perform subsampling without prefiltering, and hence suffer significant aliasing. This may be the reason, at least in part, for their performance penalty, when compared to the optimal exhaustive search.

Computational Complexity

The computational cost involved in the implementation of a MF recovery scheme may be measured in *floating point operations* (Flops). One Flop is defined here to be either an addition or a multiplication. Note that subtractions and divisions are treated as additions and multiplications respectively. The three computational costs C_{H_g} of the hierarchical Gaussian scheme, C_{H_l} of the hierarchical Laplacian scheme, and C_X of the optimal exhaustive scheme will be derived for comparison in this section. In this derivation, the video sequence consists of $W \times H$ dimensional frames Ω^n. The filters used in the pyramid generation have dimension $f \times f$. In each the MF recovery schemes, the block dimensions are $b \times b$. For the hierarchical scheme, the maximum interframe block displacement used for the block search at each level of the pyramid is denoted d_h, while for the exhaustive scheme, the maximum interframe block displacement is denoted d_x. For comparison of the two schemes, d_x is set to the maximum displacement D_N (5.1) that can be calculated from the hierarchical scheme.

To facilitate a fair comparison, the computational costs C_{H_g} and C_{H_l} of the Gaussian and Laplacian hierarchical schemes will include both the costs $C^g_{H_g}$ and $C^g_{H_l}$ of pyramid generation, and the costs $C^m_{H_g}$ and $C^m_{H_l}$ of MF recovery respectively, while the cost C_X of the optimal exhaustive scheme will include only the cost C^m_X of MF recovery as follows

$$C_{H_g} = C_{H_g}^g + C_{H_g}^m, \tag{5.3}$$

$$C_{H_l} = C_{H_l}^g + C_{H_l}^m, \tag{5.4}$$

$$C_X = C_X^m. \tag{5.5}$$

The Gaussian pyramid generation iteration consists of simply decimating the original image (see Section 3.2). On the other hand the Laplacian pyramid generation iteration consists of first decimating the original image, followed by interpolation of the decimated image and subtraction of the interpolated image from the original image (see Section 3.3). In the calculation of the computational costs of the pyramid generation iterations, the subsampling and upsampling operations are insignificant and therefore ignored. Consequently, the computational costs are solely derived from the filtering operations, plus the subtraction operation in the case of the Laplacian pyramid. In the filtering operations, a given pixel in the filtered image $\tilde{\Omega}^n$ is calculated as in (A.3). Each filtered pixel calculated as in equation A.3 requires f^2 multiplies and f^2 adds. For an $W \times H$ image $\tilde{\Omega}^n$, the total cost of a filtering operation is therefore $2f^2WH$ Flops. Clearly the cost of subtracting two images (A.2), both of dimension $W \times H$, is WH Flops. The cost of applying a single generation iteration to a $W \times H$ dimensional image in the case of the Gaussian and Laplacian pyramids is therefore $2f^2WH$ and $(4f^2+1)WH$ Flops respectively. In the generation of a N level pyramid, this iteration is applied $N-1$ times, first to the $W \times H$ original image, and then to the $\frac{W}{2} \times \frac{H}{2}$ decimated image, and so forth. The total costs C_{H_g} and C_{H_l} for generating an N level Gaussian and Laplacian pyramid are therefore given by

$$C_{H_g}^g = \frac{4}{3}(1 - 2^{-2(N-1)})2f^2WH \quad Flops, \tag{5.6}$$

and

$$C_{H_l}^g = \frac{4}{3}(1 - 2^{-2(N-1)})(4f^2+1)WH \quad Flops, \tag{5.7}$$

(see equation 4.1).

In the motion recovery schemes, the SSE (A.9) correlation measure S is used to measure the similarity between a pair of blocks B^{n-1} and B^n in adjacent frames of the video sequence. This involves b^2 subtracts, b^2 multiplies and b^2 additions, resulting in a total of $3b^2$ Flops per correlation measure. For each motion block, with a maximum interframe displacement d, this correlation measure is computed $(2d+1)^2$ times, totaling $3b^2(2d+1)^2$ Flops per motion vector. Since one motion vector is calculated for each motion block and there are $\frac{WH}{b^2}$ blocks in a $W \times H$ dimensional frame, the total cost of computing a MF is $3(2d+1)^2WH$ Flops.

For the hierarchical schemes, MF calculation is done for each of the N subimages, for total costs $C^m_{H_g}$ and $C^m_{H_l}$ of $4(1-2^{-2N})(2d_h+1)^2WH$ Flops each (see equation 4.1). Note that the cost of interpolating the MF's at various stages of the hierarchical calculation are negligible and therefore ignored in this calculation. The total costs C_{H_g} and C_{H_l} for pyramid generation and hierarchical MF recovery are therefore

$$C_{H_g} = (\frac{4}{3}(1-2^{-2(N-1)})(2f^2) + 4(1-2^{-2N})(2d_h+1)^2)WH \quad Flops, \tag{5.8}$$

and

$$C_{H_l} = (\frac{4}{3}(1-2^{-2(N-1)})(4f^2+1) + 4(1-2^{-2N})(2d_h+1)^2)WH \quad Flops. \tag{5.9}$$

These costs may be compared to the the total computational cost, C_X, for the equivalent exhaustive scheme, given by

$$C_X = 3(2(2^N-1)d_h+1)^2WH \quad Flops, \tag{5.10}$$

where d_x has been defined in terms of N and d_h (5.1).

It is of interest to compute the displacements d_h for the "break-even" points at which $C_{H_g} = C_X$ and $C_{H_l} = C_X$, since there would be no computational gain in applying the hierarchical MF recovery schemes below the breakeven point. For $f=5$, as in the case of the pyramids filters (3.2), the computational cost ratios $C_X : C_{H_g}$ and $C_X : C_{H_l}$ computed from (5.8), (5.9) and (5.10) are plotted in Figure 5.8 for various N and d_h. At the "break-even" point, the ratios are unity.

For $N=3$ level pyramids, the break-even points occur at $d_h = 0.28$ for the Gaussian pyramid, and $d_h = 0.42$ for the Laplacian pyramid. These points correspond through (4.1) to overall maximum block displacements d_x of 1.95 and 2.91 pixels respectively. On the other hand, for $N=4$ level pyramids, the break-even points occur at $d_h = 0.13$ for the Gaussian pyramid, and $d_h = 0.19$ for the Laplacian pyramid. These correspond again to $d_x = 1.95$ and $d_x = 2.92$ pixels respectively. In fact, the

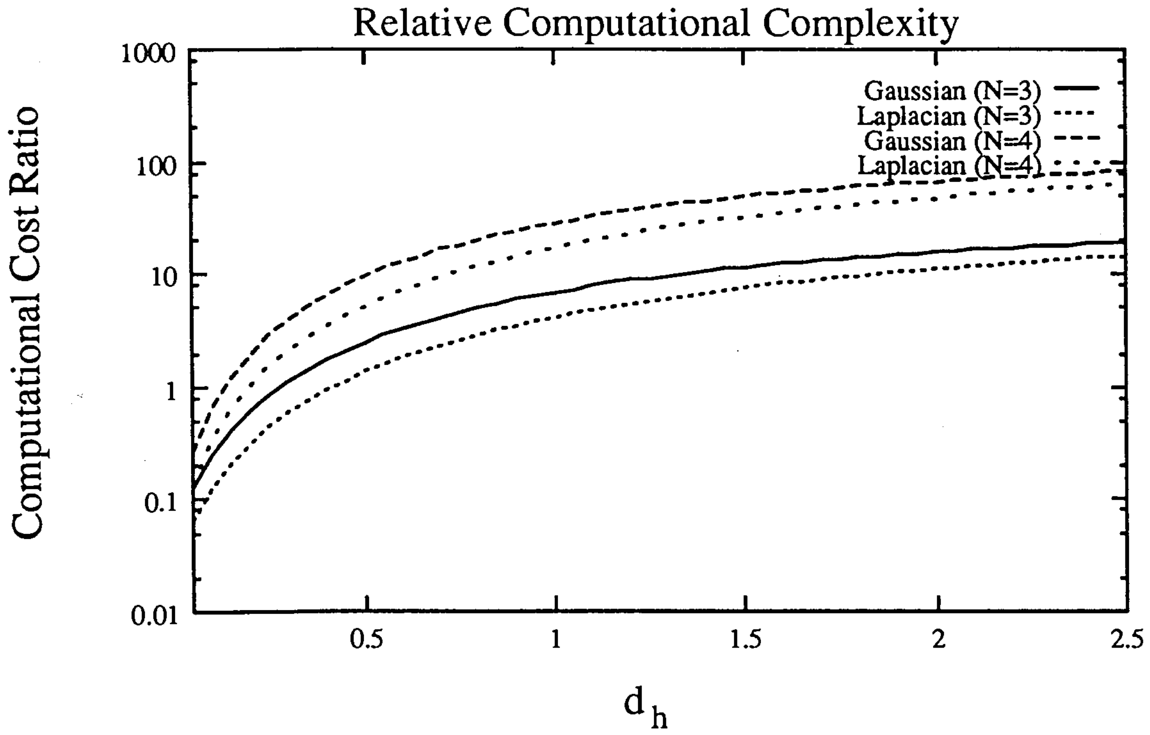


Figure 5.8: Relative Computational Cost Comparison

break-even point is practically independent of N. Therefore, for any video sequence in which the maximum interframe displacement d_x is expected to exceed one pixel, the proposed hierarchical scheme, when based on the Gaussian pyramid, will be less computationally expensive to implement than the exhaustive scheme. Similarly, for any video sequence in which the maximum block displacement is expected to exceed two pixels, the hierarchical scheme, when based on either the Gaussian or Laplacian pyramid, will be less computationally expensive to implement than the optimal exhaustive scheme. In practice, this encompasses the vast majority of video sequences.

5.3.5 Simulations

To investigate the performance of the proposed hierarchical MF recovery technique further, the simulation discussed in Section 5.3.4 was run using the "Ping Pong" video sequence (see Figure B.2). The results of the simulation show the effect of the pyramid type (Gaussian or Laplacian), the number of pyramid levels N, and the intermediate block displacement d_h on the performance of the proposed scheme.

Table 5.1: Hierarchical MF Recovery Scheme Performance Evaluation on "Ping Pong" Video Sequence

					$E_n\{E_{total}\}$			% Penalty		Comp Cost (MFlops)		
W	H	N	d_h	d_x	H_g	H_l	X	H_g	H_l	C_{H_g}	C_{H_l}	C_X
352	224	3	1	7	4.408	4.438	4.282	2.94	3.64	7.72	12.75	53.22
352	224	3	2	14	4.330	4.360	4.280	1.17	1.87	12.69	17.72	198.93
352	224	3	3	21	4.320	4.363	4.278	0.98	1.99	20.14	25.17	437.37
320	192	4	1	15	4.624	4.627	4.418	4.66	4.73	8.00	13.28	227.32

Note that, for the simulation with $N = 4$ pyramid levels, the video frames had to be cropped to ensure that the pyramid subimages had dimensions divisible by 8, the block size. The results for $N = 4$ are therefore not suitable for direct comparison with those for $N = 3$. However, it can be seen from the results that increasing the number of levels beyond $N = 3$ causes a drop in performance, in terms of percentage penalty, for the proposed scheme, whether using Gaussian or Laplacian pyramids. For the Laplacian pyramid, increasing the maximum block displacement beyond $d_h = 2$ pixels causes a drop in performance due to the unpredictable nature of the SSE correlation surface at large displacements, when based on difference subimages. These observations confirm expectations discussed in Section 5.3.1.

For all configurations, the Gaussian pyramid shows a slight performance improvement over the Laplacian pyramid in the proposed scheme. The configuration using a Gaussian pyramid with $N = 3$ pyramid levels, and a maximum block displacement of $d_h = 3$ pixels shows the best performance, with a penalty in combined entropy of only 0.98% when compared to the optimal exhaustive scheme. However, this performance is only marginally better than the 1.17% achieved by the same configuration with $d_h = 2$ pixels, at a significantly lower computational cost. Similar results were observed for the "Miss America" video sequence, where the same configuration achieved a performance penalty in combined entropy of only 1.25%. Therefore, it is proposed that the hierarchical scheme is best suited to an implementation with a $N = 3$ level Gaussian pyramid, using an intermediate block displacement of $d_h = 2$ pixels. This pyramid is best generated using a decimation filter corresponding to $\sigma_d = 0.0$.

Figure 5.9 shows the combined entropy E_{total} as a function of the frame index for the simulations presented in Table 5.1 with $N = 3$ pyramid levels. Each of the

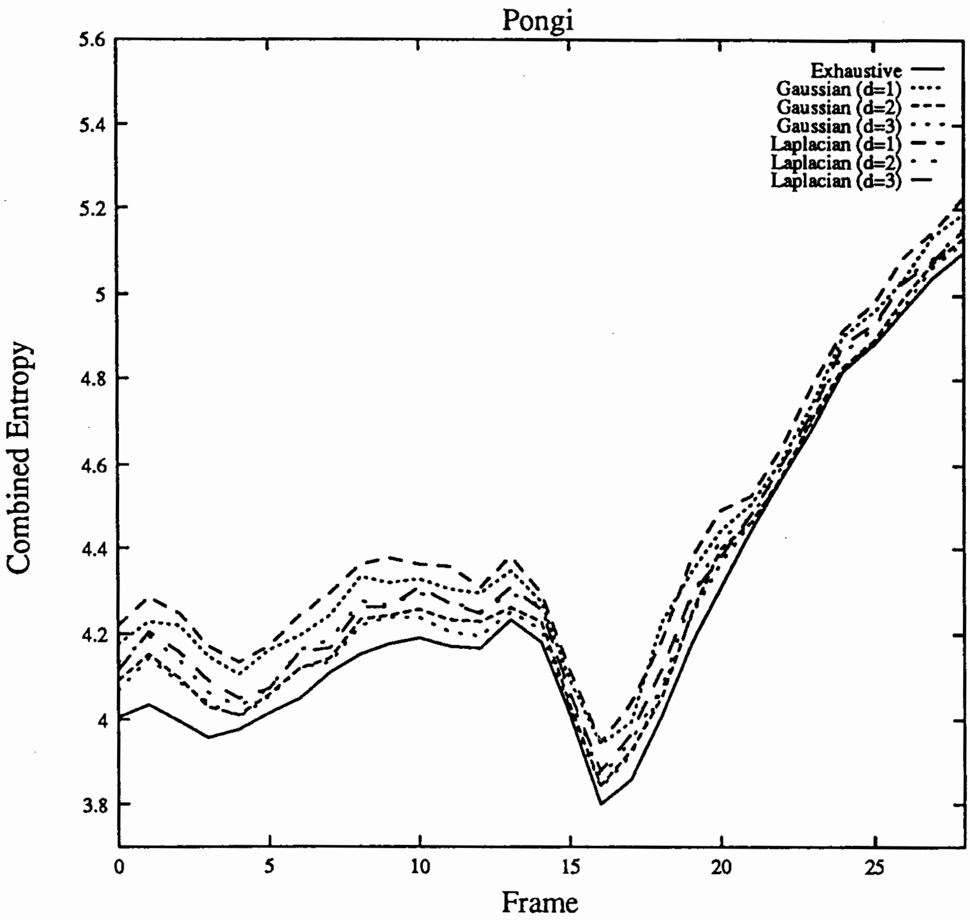


Figure 5.9: Performance of Hierarchical MF Recovery Schemes for "Ping Pong" Video Sequence

hierarchical schemes shows a close to optimal performance at all frames in the "Ping Pong" video sequence. This indicates some degree of robustness of the proposed scheme to varying types of motion in the video sequence. The $N = 3$ level Gaussian pyramid, with $d_h = 3$, again shows the best performance, although only marginally above that achieved by the same configuration with $d_h = 2$. The same performance was observed over other video sequences.

An Alternative Suboptimal MF Recovery Scheme in Comparison

The technique proposed in Section 5.3.4 is suboptimal in that it may not always find the best match motion vector for all blocks in the MF. There also exist many other suboptimal techniques that make various performance versus computational cost tradeoffs. One such suboptimal MF recovery technique, of similar computational complexity to the proposed scheme, was recently presented by Zaccarin and Liu (1992). This alternative MF recovery scheme will be investigated in order to evaluate the performance of the proposed hierarchical MF recovery technique.

Zaccarin and Liu (1992) use two strategies to reduce the computational expense of MF recovery. In the first strategy, called "Pixel Decimation", the SSE correlation measures (A.9) are calculated using only a quarter of the pixels in the blocks. Zaccarin and Liu (1992) used the *mean absolute difference* (MAD) correlation measure; however, to facilitate a fair comparison, the SSE was used for the simulation of their technique in this research. The MAD and SSE have a very similar performance in practice. To minimize the effect of the subsampling, the four alternating decimation patterns shown in Figure 5.10 are used at different positions in each search for a motion vector, as shown in Figure 5.11. Initially, a best match motion vector is computed for each decimation pattern **A**, **B**, **C** and **D**. The SSE is then recomputed, without any pixel decimation, for each of these four best match motion vectors. The motion vector amongst these four that has the lowest SSE is then chosen as the overall best match motion vector for that search.

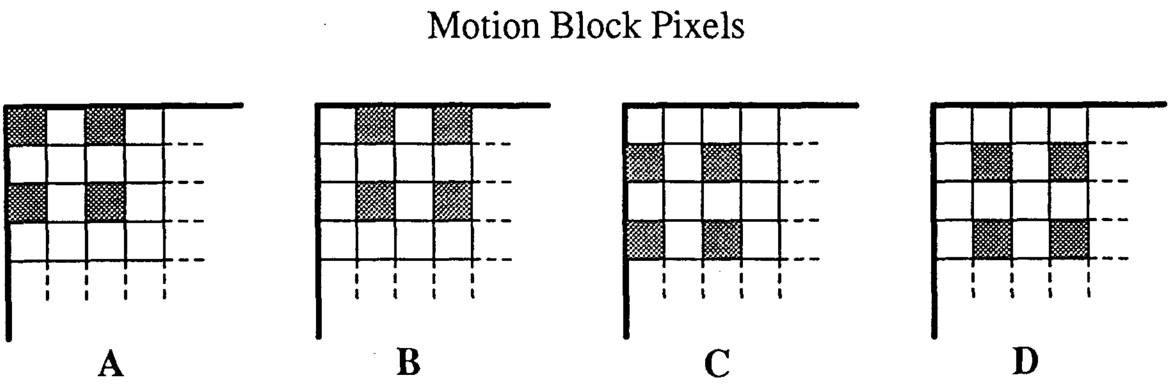


Figure 5.10: Pixel Decimation Patterns

The second strategy proposed to reduce the computational complexity of the full search, is called "Sub-block Motion Field Estimation". This initially involves computation of a quarter of the motion vectors in the subsampled MF, followed by interpolation of the full MF. Given that the optimal exhaustive search MF recovery technique subdivides the video frames into $b \times b$ motion blocks, this technique does a further subdivision into $\frac{b}{2} \times \frac{b}{2}$ motion blocks, as shown in Figure 5.12. The optimal motion

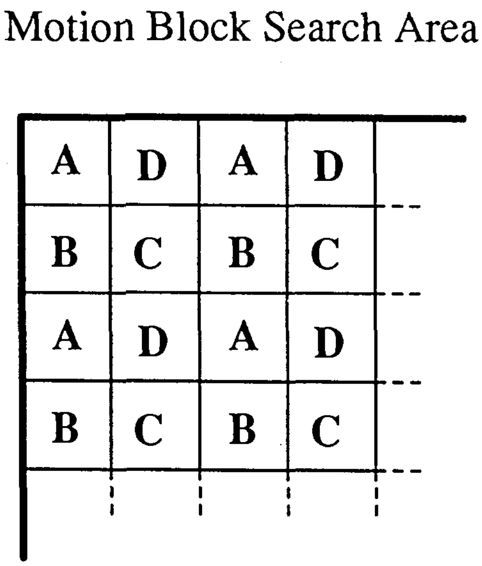


Figure 5.11: Application of the Pixel Decimation Patterns

vectors for each shaded $\frac{b}{2} \times \frac{b}{2}$ sub-block, denoted by **1** in Figure 5.12, are initially calculated using the pixel decimation scheme previously outlined. The motion vectors for the remaining sub-blocks, denoted by **2**, **3** and **4**, are then interpolated as follows. Sub-blocks **2** are assigned the motion vector of either the sub-block above or below that results in the lowest SSE match. Similarly, sub-blocks **3** are assigned the motion vector of either the sub-block to the left or right that results in the lowest SSE match. Finally, sub-blocks **4** are assigned the motion vector of the diagonally neighboring sub-block that results in the lowest SSE match. The interpolated MF can then be used by the MC encoder as described in Section 5.1. The interpolation assignments for sub-blocks **2**, **3** and **4** can be specified by 1, 1 and 2 bits respectively, resulting in a total bit overhead of 4 bits per $b \times b$ block. For most practical block sizes this overhead is small relative the the overall rate, and the full MF can be more efficiently coded as the subsampled MF and bit overhead for interpolation. The decoder can then perform the MF interpolation prior to implementation of MC decoding. The computational cost of this alternative MF recovery technique will now be developed.

Each each pixel used in the computation of the SSE correlation measure (A.9) requires 3 Flops, where a Flop is again defined as an addition or multiplication. With pixel decimation, computation of the SSE for each $\frac{b}{2} \times \frac{b}{2}$ sub-block totals $\frac{3}{16}b^2$ Flops.

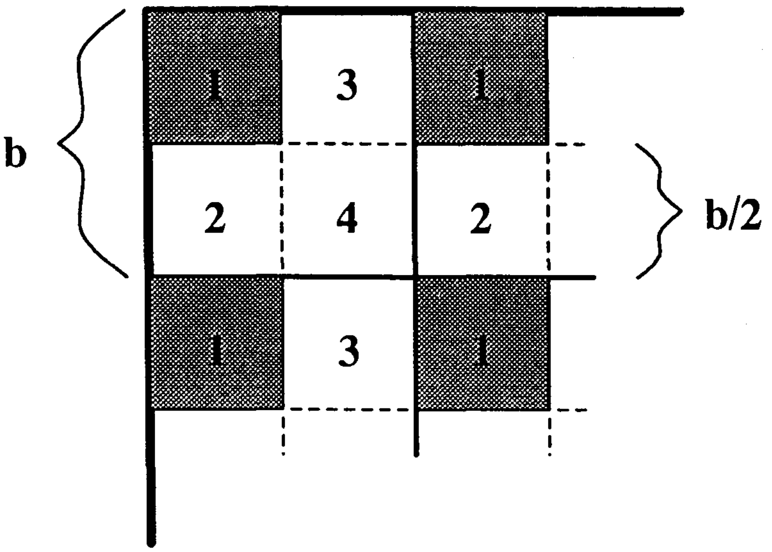


Figure 5.12: Sub-Block Motion Field Estimation

Over the search area defined by a maximum interframe block displacement d, computation of the four motion vectors based on the decimation patterns in Figure 5.10 requires $\frac{3}{16}b^2(2d+1)^2$ Flops. For each of the four resulting best match motion vectors, the SSE correlation must be recomputed without pixel decimation, adding a further $3b^2$ Flops to the computational cost per sub-block motion vector. For frames of dimension $W \times H$ pixels, $\frac{WH}{b^2}$ best match motion vectors must be computed as defined above. Interpolation of sub-blocks 2, 3 and 4 involves the computation of 2, 2 and 4 full SSE correlation measures respectively, for a total of $6b^2$ Flops per $b \times b$ block. The total computational cost C_S is therefore given by

$$C_S = (9 + \frac{3}{16}(2d+1)^2)WH \quad Flops. \tag{5.11}$$

This is approximately the same as C_{H_g} (5.8), the computational cost required by the proposed hierarchical MF recovery technique when implemented based on an $N = 3$ level Gaussian pyramid and maximum intermediate block displacement $d_h = 2$ pixels. These suboptimal techniques require approximately $\frac{1}{16}$ the computational cost of the full search technique C_X (5.10). Simulations were done to evaluate the performance of this alternative MF recovery scheme against that of both the proposed hierarchical scheme, and the optimal exhaustive scheme.

To evaluate the performance of these various MF recovery techniques, simulations were run on both the "Miss America" and "Ping Pong" sequences (see Figure B.2).

In each of the MF recovery techniques, a block size of 8 × 8 pixels was used. This corresponded to a sub-block size of 4 × 4 pixels in the scheme proposed by Zaccarin and Liu (1992). The hierarchical scheme used $N = 3$ level Gaussian pyramids and an intermediate maximum block displacement $d_h = 2$ pixels. Equivalently, a maximum block displacement of 14 pixels was used in both the exhaustive scheme, and the scheme proposed by Zaccarin and Liu (1992). Figures 5.13 and 5.14 show the combined entropy E_{total} (5.2) of the MF and DFD for each frame of each sequence, resulting from the use of each of the MF recovery schemes. These results are also summarized for comparison in Table 5.2. Note that the overhead information of 4 bits per 8 × 8 block required to interpolate the MF in the technique proposed by Zaccarin and Liu (1992) is included in the combined entropy, in bits per pixel, shown in the table.

Table 5.2: Summary of Performance Evaluation of MF Recovery Schemes

Video Sequence			Cost (MFlops)			$E_n\{E_{total}\}$ (bpp)			Penalty (%)	
Name	W	H	C_S	C_{H_g}	C_X	S	H	X	S	H
Miss America	352	288	16.29	16.32	255.77	3.65	3.38	3.34	-9.36	-1.20
Ping Pong	352	224	12.67	12.69	198.93	4.62	4.33	4.28	-8.00	-1.17

Clearly, the proposed hierarchical scheme outperforms the alternative scheme by a significant margin for both sequences, while at approximately the same computational cost.

The Effect of Quantization Noise on Hierarchical MF Recovery

In a practical MC codec, the DFD would generally be quantized at the encoder. This introduces distortion into the frames reconstructed by the MC decoder. Since the objective in the design of the MC codec is to minimize the average distortion of the target frames reconstructed by the MC decoder, the MC encoding should be based on the source frame available at the decoder that contains the quantization noise. Simulations were done to investigate the effect of quantization noise in the source frame on the proposed hierarchical MF recovery scheme for the case of both the Gaussian and Laplacian pyramids. Figure 5.15 outlines the data flow in one iteration

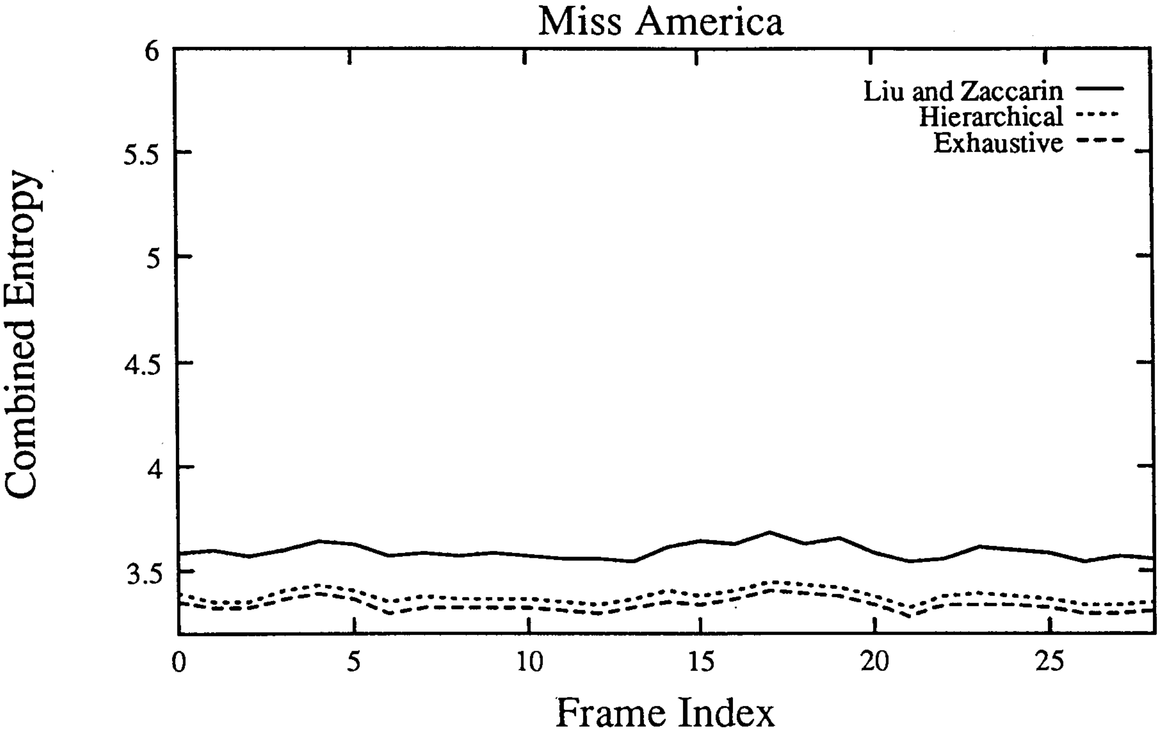


Figure 5.13: Performance Comparison for MF Recovery Schemes on "Miss America" Video Sequence

of the simulation, which was performed on each pair of neighboring source and target frames in the "Ping Pong" and "Miss America" video sequences.

After quantization of the source frame, three level pyramids were generated from the source and target frames. The MF was then recovered hierarchically from the pyramids, using an 8 × 8 block size and intermediate maximum block displacement $d_h = 2$ pixels. The quantized source frame, target frame and MF were then used by the MC encoder to generate the DFD. After calculating the entropies of both the MF and DFD, the combined entropy (5.2) was calculated. Figure 5.16 shows the combined entropy averaged over the frame sequences for the two types of pyramids and various quantizer rates. For comparison, the average combined entropy achieved by the optimal exhaustive scheme, based on an 8 × 8 block size and maximum displacement D_N of 14 pixels (see equation 5.1), is shown for comparison, both with and without quantization of the source frame.

Note that the exhaustive MF recovery technique, without quantization of the

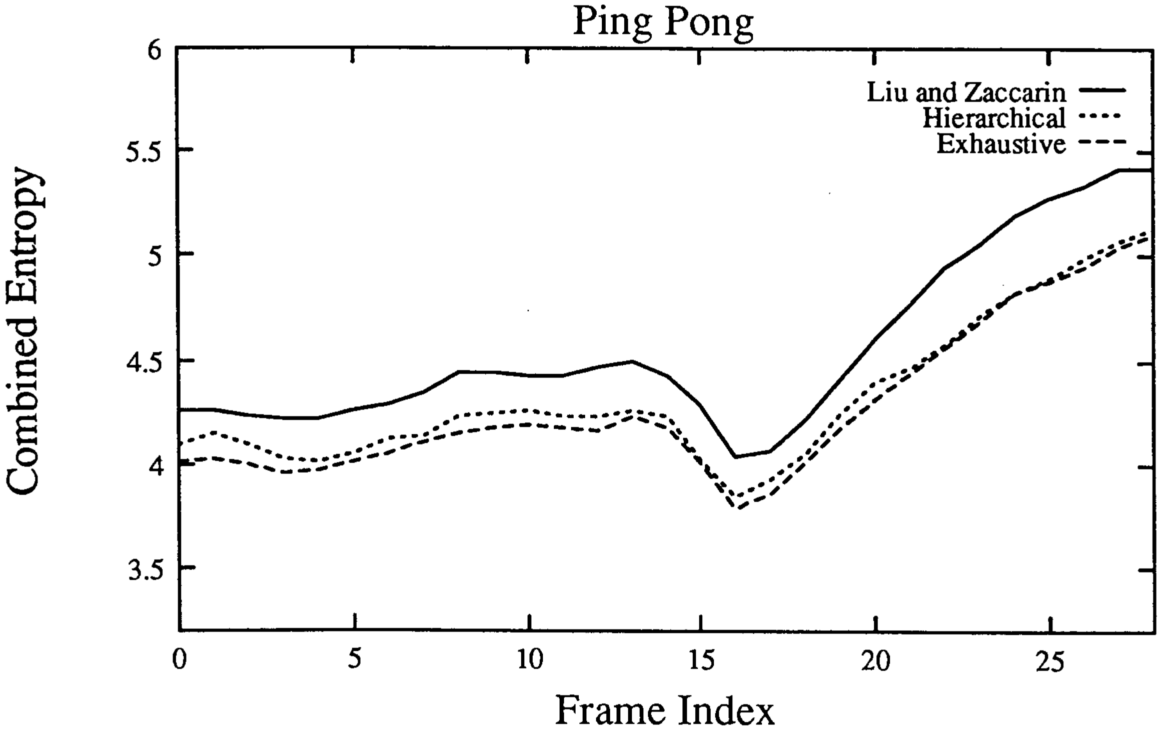


Figure 5.14: Performance Comparison for MF Recovery Schemes on "Ping Pong" Video Sequence

source frame, achieves the best performance in terms of minimizing the average combined entropy, as expected. The exhaustive MF recovery scheme, with quantization of the source frame, achieves a similar performance at high quantizers rates, but its performance drops at lower rates where quantization noise in the source frame has a significant effect on the MC coder, increasing the entropy of both the MF and DFD. For all quantizer rates, hierarchical MF recovery, with either the Gaussian or Laplacian pyramids, achieves a performance close to that of the optimal exhaustive technique, with quantization of the source frame.

On average, hierarchical MF recovery performs slightly better when based on the Gaussian pyramid, than when based on the Laplacian pyramid. This is probably due to the fact that the SSE correlation surface, obtained while searching for a motion vector at a given stage in the hierarchical MF recovery scheme, is better behaved in the case of the Gaussian pyramid, and is therefore more robust to quantization noise in the source frame for example. Thus, in the case of the Gaussian pyramid, the hierarchically recovered MF will contain less bad matches, and will be closer to

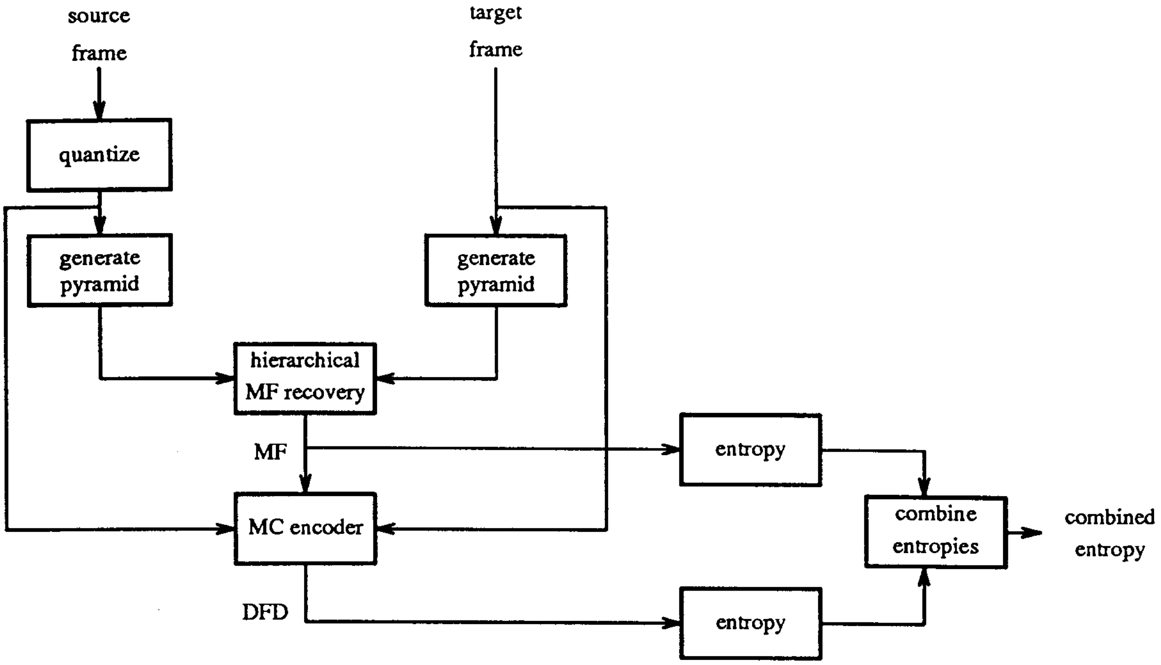


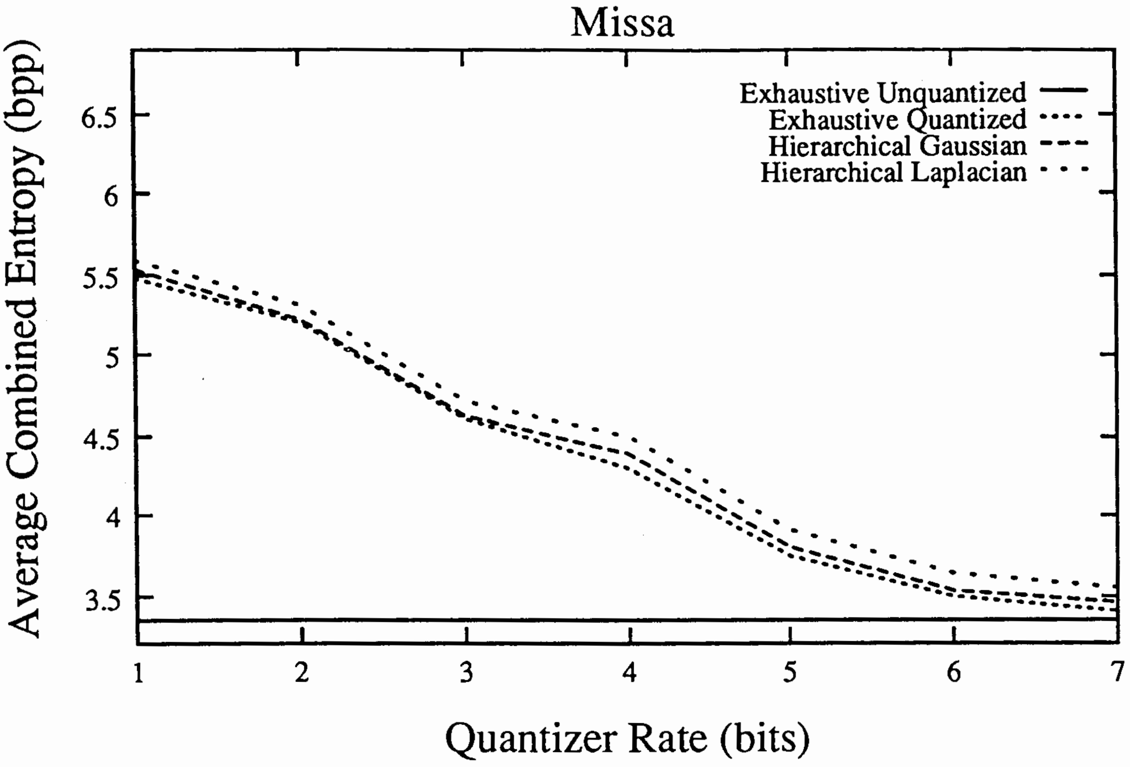
Figure 5.15: Simulation Data Flow Diagram

optimal than in the case of the Laplacian pyramid.

For some quantizer rates, the hierarchical Gaussian MF recovery scheme performs marginally better than the exhaustive scheme, with quantization of the source frame. It is proposed that this unusual result is due to the fact that the exhaustive scheme is optimal only in the sense that it finds the best match in terms of minimizing the energy of the DFD. However, with quantization of the source frame the entropy of the MF recovered by the exhaustive technique, and therefore the combined entropy, may increase significantly as a result of the decreasing quality of the block matches. On the other hand, hierarchical MF recovery techniques lead to more accurate motion fields in the sense of the true motion (Dufaux and Kunt 1992), making them more robust to such increases in the entropy of the recovered MF as a result of quantization of the source frame.

Hierarchical MF Recovery In A Practical Codec

The hierarchical scheme for MF recovery outlined in Section 5.3.4 was evaluated both in Sections 5.3.5 in terms of relative performance, and in Section 5.3.5 in terms of robustness to quantization noise in the source frame. It is also useful to investigate



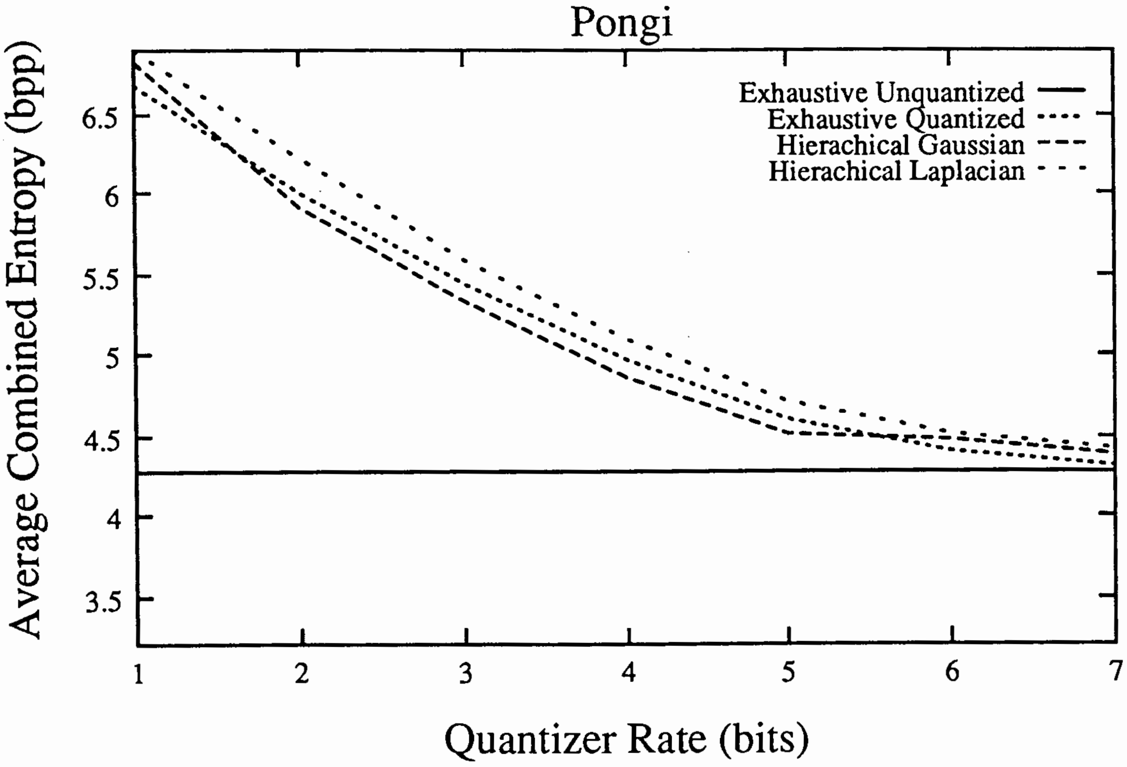


Figure 5.16: Average Combined Entropy Simulation Results

the performance of the proposed scheme in a real codec implementation.

Figure 5.17 shows a block diagram of an MC codec employing hierarchical MF recovery. $\mathbf{Q}$ and $\mathbf{Q}^{-1}$ represent quantization and inverse quantization, respectively, of the DFD. Similarly, $\mathbf{E}$ and $\mathbf{E}^{-1}$ represent entropy coding and decoding, respectively, of the MF and DFD. Finally, $\mathbf{D}$ represents a single frame delay. It is desirable in the codec to maximize the quality of the reconstructed target frame $\hat{\Omega}_t$ output from the decoder, when compared to the original target frame Ω_t input to the coder. Note that, in each iteration of the codec, the previously reconstructed target frame $\hat{\Omega}_t$ is used as the source frame $\hat{\Omega}_s$ for the current MC coding operation (see Section 5.1). Therefore, the source frame includes the effects of DFD quantization in the MC coder. Note that in a real implementation, the initial source frame $\hat{\Omega}_s$ would be encoded without MC, as previously discussed. The hierarchical MF recovery was based on three level Gaussian pyramids created using a decimation filter with $\sigma_d = 0.0$, an 8×8 block size, and a maximum intermediate displacement d_h of two pixels.

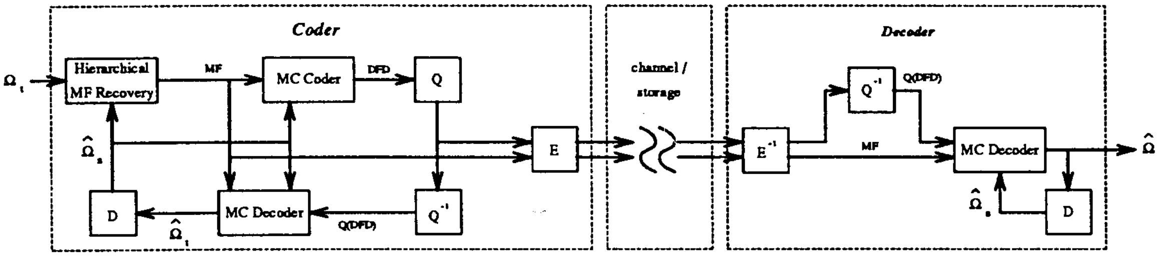


Figure 5.17: A Practical MC Codec Employing Hierarchical MF Recovery

The performance of the codec was evaluated using both the "Miss America" and "Ping Pong" sequences (see Figure B.2). For each iteration of the codec, the PSNR (A.13) of the reconstructed target frame $\hat{\Omega}_t$ was measured, when compared to the original target frame Ω_t. Similarly, the rate was measured for each reconstructed target frame $\hat{\Omega}_t$ as the combined entropy (5.2) of the MF and DFD. Figure 5.18 shows the average PSNR at different average rates. Note that the different rates were achieved by varying the bits allocated to the DFD quantizer.

At low rates, the PSNR increases approximately linearly with increasing rate, indicating a logarithmic decrease in the variance of the noise introduced through DFD quantization. The average PSNR is higher at a given rate for the "Miss America" sequence, since on average it contains less motion than in the "Ping Pong" sequence.

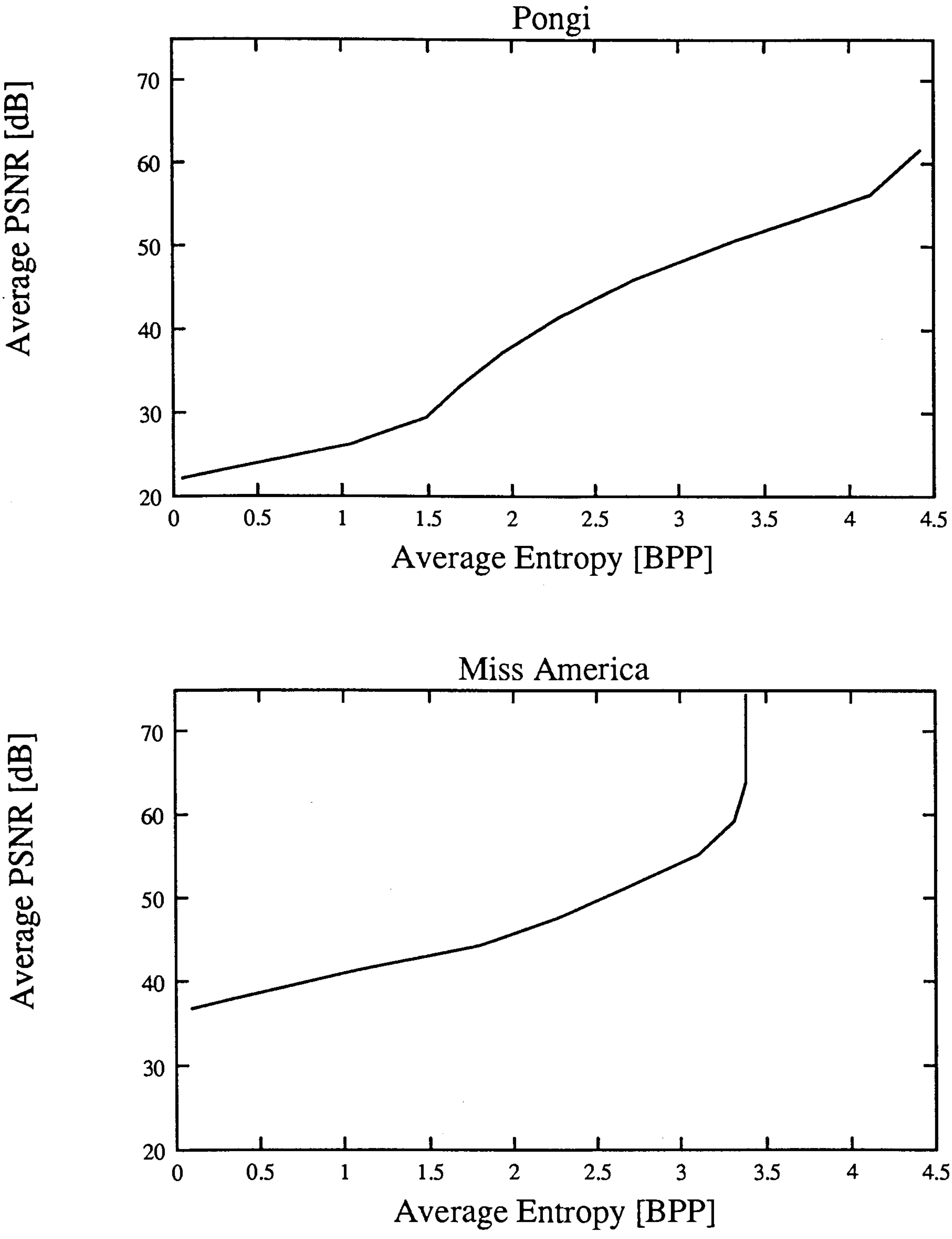


Figure 5.18: Performance of MC Codec Using Hierarchical MF Recovery

This allows the MC coder to interpolate the target frame accurately, and leads to a low energy DFD that may be efficiently quantized at lower rates. In fact, for higher rates in the "Miss America" simulation results, quantization of the DFD has almost no effect, and the quality of the reconstructed sequence is nearly perfect. Generally, the quality of the reconstructed video signal is reasonable at an average PSNR of approximately 30.0dB. For the "Miss America" sequence, an average PSNR of 36.9dB is achieved even with no bits allocated to the DFD quantizer, at a rate of only 0.09 bpp, corresponding to a compression ratio (A.14) of 90.9. However, for the higher motion "Ping Pong" sequence, a PSNR of 29.5dB is achieved only at a rate of 1.49 bpp, corresponding to a compression ratio of 5.4.

Note that these results could be improved with the use of more elaborate schemes in the source coding of the MF and DFD, for example DPCM, subband coding or pyramid coding. Furthermore, quantization schemes such as vector quantization or the DCT could be used, to exploit remaining interpixel correlation in the DFD, to achieve a higher coding gain.

Chapter 6

Conclusion

Pyramid coding is a technique used to reduce information redundancy in a source, thereby improving storage or transmission efficiency. Feedback in the pyramid generation process allows both flexibility in the choice of generation filters, and the use of quantization feedback. This in turn allows pyramid coding to be optimized for a particular application; for example, either directly in lossless or lossy compression, or indirectly in the efficient recovery of motion fields from video sequences.

For applications that require perfect reconstruction of the original image, lossless pyramid coding may be used. For example, legal specifications require that medical images maintain perfect quality through the source codec. Alternatively, higher image compression ratios may be achieved using lossy pyramid coding, for example in applications such as image databasing. The hierarchical structure of the image pyramid makes it suitable for both the progressive transmission of images over slow channels, for example in remote searching of image databases, as well as the efficient recovery of motion information from video sequences, for example in temporal video source coding.

This thesis has outlined the principles of pyramid coding and explored new ways in which to optimize its performance in various applications. In particular, a new class of filters that are well suited to pyramid coding have been developed. It has also been demonstrated that the choice of these filters has a significant effect on the performance of the pyramid codec in a broad range of applications. An evaluation of this performance has been presented, both in relation to optimal methods, and to other suboptimal techniques. Results show that lossless pyramid coding, or more specifically MEP coding, performs particularly well relative to other techniques, both in terms of higher compression ratios and lower computational cost. Furthermore, pyramid coding achieves very near optimal performance in the recovery of motion fields from video sequences, at only a fraction of the computational cost. On the other hand, due to quantization feedback in the pyramid codec, the estimation of distortion in the image reconstructed from the quantized pyramid is a highly nonlinear problem, and complicates the task of bit allocation. Therefore, although the evaluation of lossy pyramid coding presented in this thesis shows promising results, further research is required to find more reliable bit allocation methods. Similarly, the subjective quality of an image reconstructed from a quantized pyramid may differ significantly from the objective quality, measured for example by the PSNR, due to various characteristics of the HVS. A detailed study of these subjective properties of the HVS would lead to better choices of filters for lossy pyramid coding of a particular image. However, these topics are beyond the scope of this thesis and are recommended for future research.

Bibliography

Akansu, A. (1992). *Multiresolution Signal Decomposition : transforms, subbands, and wavelets.* Boston: Academic Press.

Akansu, A., J. Chien, and M. Kadur (1989). Lossless compression of block motion information in motion compensated video coding. *Proceedings SPIE, Visual Communications and Image Processing 1199*, 30–38.

Anandan, P. (1987, May). A unified perspective on computational techniques for the measurement of visual motion. *International Conference on Computer Vision* , 219–230.

Barnsley, M. F. and L. P. Hurd (1993). *Fractal Image Compression.* Wellesley, Massachusetts: AK Peters Ltd.

Bell, T., J. Cleary, and I. Witten (1990). *Text Compression.* Prentice Hall.

Bergen, J. and E. Adelson (1987). Hierarchical, computationally efficient motion estimation algorithm. *Journal of the Optical Society of America, A 4*, P35.

Bergen, J. and P. Burt (1990). Computing two motions from three frames. *IEEE Proceedings of the International Conference on Computer Vision* , 27–32.

Bierling, M. (1988). Displacement estimation by hierarchical blockmatching. *Proceedings SPIE, Visual Communications and Image Processing 1001*, 942–950.

Burt, P. and E. Adelson (1983, April). The Laplacian pyramid as a compact image code. *IEEE Transactions on Communications COM-31*, 532–540.

Burt, P., R. Hingorani, and R. Kolczynski (1991). Mechanisms for isolating component patterns in the sequential analysis of multiple motion. *IEEE Workshop on Visual Motion* , 187–193.

Burt, P., C. Yen, and X. Xu (1982). Local correlation measures for motion analysis, a comparative study. *IEEE Proceedings PRIP* , 269–274.

Chen, Y., K. Sayood, and D. Nelson (1992). A robust coding scheme for packet video. *IEEE Transactions on Communications* , 1491–1501.

Chun, K. and J. Ra (1992). Fast block matching algorithm by successive refinement of matching criterion. *Proceedings SPIE, Visual Communications and Image Processing* , 552–560.

Clarke, R. (1985). *Transform Coding of Images.* New York: Academic Press.

Dengler, J. (1986). Local motion estimation with the dynamic pyramid. In V.Cantoni and S. Levialdi (Eds.), *Pyramidal Systems for Computer Vision.* Springer-Verlag.

Dufaux, F. and M. Kunt (1992). Multigrid block matching motion estimation with an adaptive local mesh refinement. *Proceedings SPIE, Visual Communications and Image Processing* , 97–109.

Elliot, D. (1987). *Handbook of Digital Signal Processing : Engineering Applications.* San Diego: Academic Press.

Esteban, D. and C. Galand (1977, May). Application of quadrature mirror filters to split band voice coding schemes. In *Proceedings ICASSP 77*, pp. 191–195. IEEE.

Farvardin, N. and J. W. Modestino (1984, May). Optimum quantizer performance for a class of non-gaussian memoryless sources. *IEEE Transactions on Information Theory 30*(3), 485–497.

Furlan, G. (1991, May). An enhancement to universal modelling algorithm context for real-time applications to image compression. In *Proceedings ICASSP'91*, pp. 2777–2781.

Gandhi, R., L. Wang, S. Panchanathan, and M. Goldberg (1993, November). An mpeg-like pyramidal video coder. *Proceedings SPIE, Visual Communications and Image Processing 2094*(2), 706–717.

Gersho, A. and R. Gray (1992). *Vector Quantization and Signal Compression.* Series in Communications and Information Theory. Kluwer Academic Publishers.

Glazer, F., G. Reynolds, and P. Anandan (1983, June). Scene matching by hierarchical correlation. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* , 432–441.

Goldberg, M. and L. Wang (1991, April). Comparative performance of pyramid data structures for progressive image transmission. *IEEE Transactions on Communications 39*(4), 540–548.

Gothe, M. (1993, July). *On The Integration of Motion Compensation With Subband Filter Coding Techniques.* M.a.sc. thesis, Simon Fraser University.

Gurski, G., M. Orchard, and A. Hull (1992). Optimal linear filters for pyramidal decomposition. In *Proceedings ICASSP'92*, Volume 4, pp. IV–633– IV–636.

Ho, Y. and A. Gersho (1989a). Classified transform coding of images using vector quantization. In *IEEE Proceedings of ICASSP.*

Ho, Y. and A. Gersho (1989b). A pyramidal image coder with contour-based interpolative vector quantization. *Proceedings SPIE, Visual Communications and Image Processing 1199*, 733–740.

Houlding, D. and J. Vaisey (1994, February). Low entropy image pyramids for efficient lossless coding and progressive transmission. *Proceedings SPIE Conference on Image and Video Compression 2186*, 88–97.

Hsing, T. R. (1987). Motion detection and compensation coding for motion video coders: Technical review and comparison. In *GLOBECOM '87*, pp. 2.6.1–2.6.4.

Huffman, D. A. (1952, September). A method for the construction of minimum redundancy codes. *Proceedings of the IRE 40*, 1098–1101.

Jayant, N. and P. Noll (1984). *Digital Coding of Waveforms, Principles and Applications to Speech and Audio.* Prentice Hall.

Koga, T., K. Iinuma, A. Hirano, and Y. Iijima (1981). Motion compensated interframe coding for video conferencing. *NTC' 81 Conference Record* , G5.3.1–G5.3.5.

Kronander, T. (1989, January). *Some Aspects of Perception Based Image Coding.* Ph.D dissertation, Linköping University, Linköping Sweden S-581 83.

Langdon, G. G. and J. Rissanen (1981, June). Compression of black-white images with arithmetic coding. *IEEE Transactions on Communications 29*(6), 858–867.

Lin, S. and D. Costello (1983). *Error Control Coding : Fundamentals and Applications.* Prentice-Hall.

Marr, D. (1982). *Vision.* W. H. Freeman and Company.

Netravali, A. and B. Haskell (1988). *Digital Pictures, Representation and Compression.* Applications of Communication Theory. Plenum.

Oppenheim, A. V. and R. W. Schafer (1989). *Discrete-Time Signal Processing.* Prentice Hall.

Pennebaker, W. and J. Mitchell (1993). *JPEG : Still Image Data Compression Standard.* New York: Van Nostrand Reinhold.

Press, W. H. (1992). *Numerical Recipes in C : The Art of Scientific Computing.* New York: Cambridge University Press.

Riskin, E. (1991, March). Optimal bit allocation via the generalized BFOS algorithm. *IEEE Transactions on Information Theory 37*(2), 400–402.

Sezan, M. and R. Lagendijk (1993). *Motion Analysis and Image Sequence Processing.* Kluwer Academic Publishers.

Singh, A. (1991). *Optic Flow Computation : A Unified Perspective.* IEEE Computer Society Press.

Smith, M. and S. Eddins (1987). Subband coding of images with octave band tree structures. In *Proceedings ICASSP'87*, pp. 1382–1385.

Stiller, C. and D. Lappe (1991). Laplacian pyramid coding of prediction error images. *Proceedings SPIE, Visual Communications and Image Processing 1605*, 47–57.

Torbey, H. and H. Meadows (1989). System for lossless digital image coding / decoding. *Proceedings SPIE, Visual Communications and Image Processing 1199*, 989–1002.

Uz, K., M. Vetterli, and D. LeGall (1991, March). Interpolative multiresolution coding of advanced television with compatible subchannels. *IEEE Transactions on Circuits and Systems for Video Technology 1*(1), 86–99.

Wang, L. and M. Goldberg (1989a, December). Progressive image transmission using vector quantization on images in pyramid form. *IEEE Transactions on Communications 37*(12), 1339–1349.

Wang, L. and M. Goldberg (1989b, July). Reduced-difference pyramid : a data structure for progressive image transmission. *Optical Engineering 28*(7), 708–716.

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. (1991). *Subband Image Coding.* Boston: Kluwer Academic Publishers.

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.

Ziv, J. and A. Lempel (1978). Compression of individual sequences via variable rate encoding. *IEEE Transactions on Information Theory 24*, 530–536.

Appendix A

Mathematical Formulae

Let an image Ω be referenced using a two dimensional Cartesian grid so that a given pixel is specified by its (i,j) coordinates, and denoted as Ω_{ij}. The top left pixel is assigned the coordinates $(0,0)$ with i and j coordinates increasing to the right and down the image respectively. The image width and height are represented by w and h respectively.

A.1 General Image Operations

A.1.1 Addition

The summed image Ω^s is calculated by the pixel-wise addition of Ω^1 and Ω^2 as follows

$$\Omega^s = \sum_{j=0}^{h-1} \sum_{i=0}^{w-1} \Omega^1_{i,j} + \Omega^2_{i,j}. \tag{A.1}$$

A.1.2 Subtraction

Similarly, the difference image Ω^d is calculated by the pixel-wise subtraction of Ω^1 and Ω^2 as follows

$$\Omega^d = \sum_{j=0}^{h-1} \sum_{i=0}^{w-1} \Omega^1_{i,j} - \Omega^2_{i,j}. \tag{A.2}$$

A.1.3 Filtering

A two dimensional $f \times f$ FIR filter F, with its center coefficient indexed at $(0,0)$, may be used to filter the image Ω^n to give $\tilde{\Omega}^n$ as follows

$$\tilde{\Omega}^n = \sum_{j=0}^{h-1} \sum_{i=0}^{w-1} \sum_{k=-f/2}^{f/2} \sum_{l=-f/2}^{f/2} F_{k,l}\, \Omega^n_{i+k,j+l} \tag{A.3}$$

A.2 General Image Measures

A.2.1 Rate (R) [bits per pixel]

$$R = \frac{1}{w * h} \sum_{j=0}^{h} \sum_{i=0}^{w} r_{ij} \tag{A.4}$$

Where

r_{ij} = bits allocated to represent pixel (i, j).

A.3 Image Statistics Measures

A.3.1 Mean (μ)

$$\mu = \frac{1}{w * h} \sum_{j=0}^{h} \sum_{i=0}^{w} \Omega_{ij} \tag{A.5}$$

A.3.2 Variance (σ^2)

$$\sigma^2 = \frac{1}{w * h} \sum_{j=0}^{h} \sum_{i=0}^{w} (\Omega_{ij} - \mu)^2 \tag{A.6}$$

A.3.3 Standard Deviation (σ)

$$\sigma = \sqrt{\sigma^2} \tag{A.7}$$

A.3.4 Entropy (H) [bits per symbol]

The zeroth order entropy H is calculated as

$$H = \sum_{a \epsilon A} p_a \log_2 \frac{1}{p_a} \tag{A.8}$$

Where

$$\begin{aligned} A &= \text{the complete set of image pixel values.} \\ a &= \text{an element of the set } A. \\ p_a &= \text{the probability of } a \text{ in } A. \end{aligned}$$

A.3.5 Sum of Squared Error (SSE) Correlation Measure

For a motion block $B^n_{k,l}$ in frame n, referenced by its upper left pixel at coordinates (k, l), the SSE correlation measure $S_{i,j}$ with a block in the previous frame, displaced by the vector (i, j), is given by

$$S_{i,j} = \sum_{k=1}^{b} \sum_{l=1}^{b} (B^{n-1}_{k+i,l+j} - B^n_{k,l})^2. \tag{A.9}$$

where b represents the width and height of the blocks.

A.4 Image Quality Measures

Let the original image be denoted Ω_{ij}, and the reconstructed image (the original image with added distortion) be denoted Ω'_{ij}.

A.4.1 Mean Squared Error (MSE)

$$MSE = \frac{1}{w * h} \sum_{j=0}^{h} \sum_{i=0}^{w} (\Omega_{ij} - \Omega'_{ij})^2 \tag{A.10}$$

A.4.2 Normalized Mean Squared Error (NMSE)

$$NMSE = \frac{\sum_{j=0}^{h} \sum_{i=0}^{w} (\Omega_{ij} - \Omega'_{ij})^2}{\sum_{j=0}^{h} \sum_{i=0}^{w} (\Omega_{ij})^2} \tag{A.11}$$

A.4.3 Signal to Noise Ratio (SNR) [dB]

$$SNR = 10\log_{10}\frac{\sigma^2}{MSE} \tag{A.12}$$

A.4.4 Peak Signal to Noise Ratio (PSNR) [dB]

$$PSNR = 10\log_{10}\frac{2^{2n}}{MSE} \tag{A.13}$$

Where

n = bits per pixel in original image.

A.5 Compression Measures

A.5.1 Compression Ratio (C)

$$C = \frac{R}{R_c} \tag{A.14}$$

Where

R = rate of the original image.

R_c = rate of the compressed image.

Appendix B

Standards

B.1 Source Coding Standards

The objective in source coding is to facilitate efficient communication and storage of data (see Chapter 1). However, there exist many diverse techniques for the compression of images and video. In order to facilitate the exchange of compressed images or video, standards have been formulated. Since these standards effectively determine the use of various source coding techniques, a brief outline will now be given. Pennebaker and Mitchell (1993) provide a more detailed description of these standards.

B.1.1 CCITT

International Telegraph and Telephone Consultative Committee set two standards, G3 and G4, for binary image compression. These standards are, at present, used for fax compression. However, they are likely to be replaced by the JBIG standard.

B.1.2 JBIG

Joint Bi-level Image Experts Group (JBIG) defines a new standard for lossless compression of fax images.

B.1.3 JPEG

Joint Photographic Experts Group (JPEG) is an international standard for color image compression. The JPEG group formulated the standard under the supervision of three major international standards organizations, namely the *International Organization for Standardization* (ISO), the *International Telegraph and Telephone Consultative Committee* (CCITT), and the *International Electrotechnical Commission* (IEC), in order to facilitate the communication of compressed images. JPEG describes an architecture incorporating a set of image source coding methods that make it a suitable standard for a wide range of image compression applications. A form of Laplacian pyramid coding (see Section 3) has been implemented in the JPEG "hierarchical mode", where progressive transmission is facilitated.

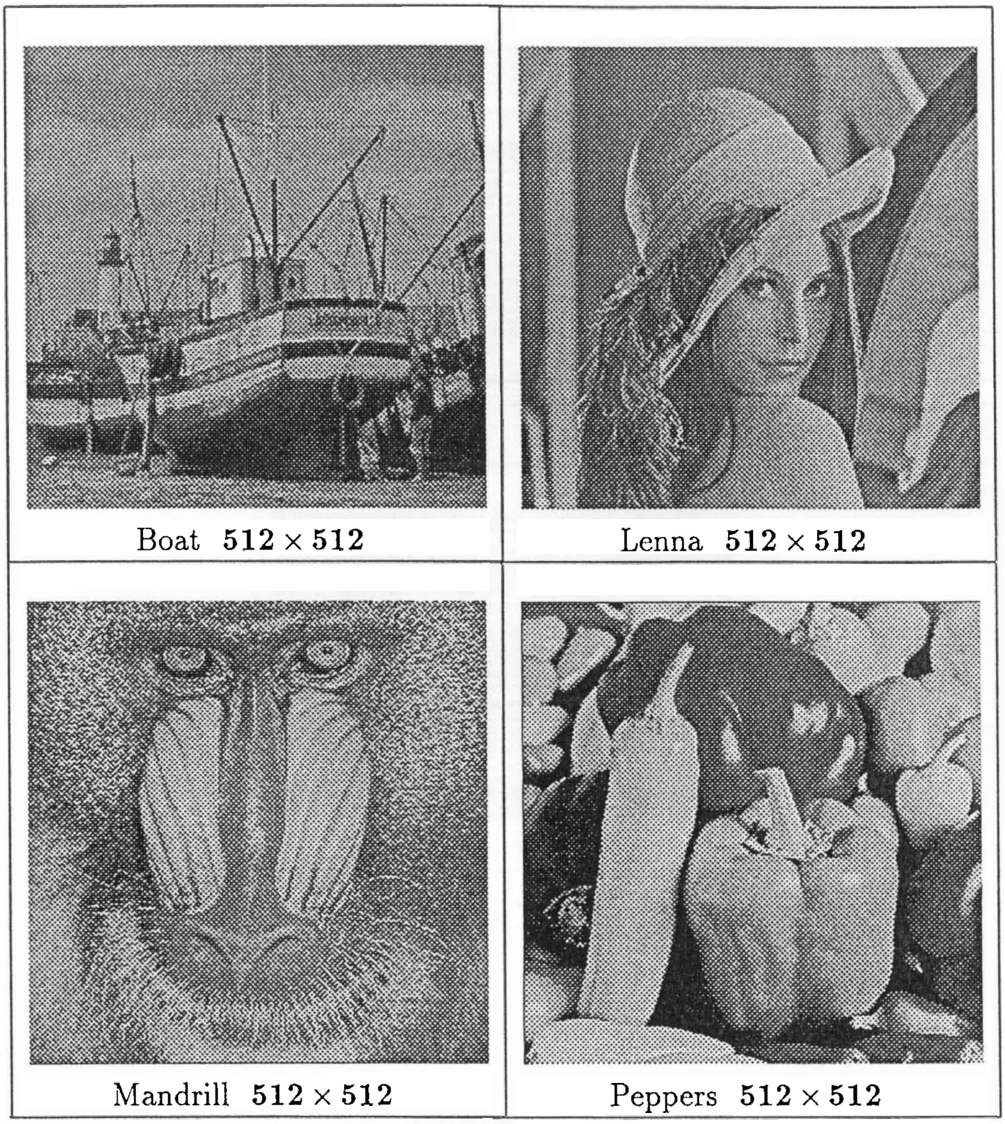
B.1.4 H.261

This is a CCITT motion sequence compression standard for low bandwidth, real time video compression. This standard is also known as $P \times 64$, since it is intended for use, in teleconferencing applications, at rates that are multiples of 64 kBPS. The H.261 standard is to be replaced by the newer MPEG standard.

B.1.5 MPEG

Moving Picture Experts Group (MPEG) defines a standard for the compression of video sequences. This standard had two phases of development. The first phase, named MPEG-1, was intended for resolutions of 320×240 and bit rates of approximately 1.5 MBPS, for both video and two audio channels. Subsequently, MPEG-2 was developed for higher resolutions and rates of approximately 4-10 MBPS.

B.2 Standard Test Images



Boat 512×512

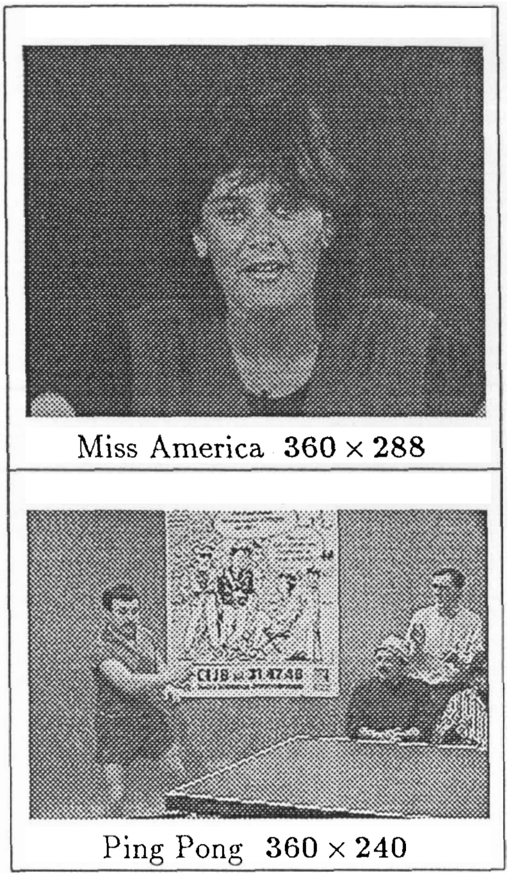
Lenna 512×512

Mandrill 512×512

Peppers 512×512

Figure B.1: Test Images (8 bpp)

B.3 Standard Test Video Sequences



Miss America 360 × 288

Ping Pong 360 × 240

Figure B.2: Test Videos (8 bpp)