STILL AND MOVING IMAGE COMPRESSION SYSTEMS USING MULTISCALE TECHNIQUES

Thesis by

Ming-Chieh Lee

In Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy

California Institute of Technology
Pasadena, California
1994

(Submitted December 2, 1993)

Acknowledgement

First of all, I would like to dedicate this thesis to Professor Edward C. Posner (1933–1993). I was very fortunate to have him as my advisor. His enthusiasm in helping and supporting students without any reward, his attitude to doing research and all other things, and his constant encouragement made a great impact on my life. Without his generosity and kindness, I would not have had such a smooth and meaningful graduate life at Caltech. He always advised me that, to be a good Ph.D. candidate, working on the research as independently as possible is necessary. I have been trying hard to follow his direction, and will do so in my future career, although I know I can never achieve his expectation. I really learned a lot from this intellectual and generous person. It is almost impossible for me to express my gratitude to him in such limited space. His death is certainly an unimaginable loss to science, his family, friends, and students.

I also want to thank Professor Robert J. McEliece for taking charge of my research work after Dr. Posner's accident. Without his generous assistance and tremendous effort (we made him extremely busy), my graduate study at Caltech would have been interrupted.

Thanks go to Doctors Kar-Ming Cheung, Robert J. McEliece, Carver A. Mead, Fabrizio Pollara, Marvin K. Simon, and P. P. Vaidyanathan for their interest in my work and for their valuable comments, also for their time serving on my candidacy and/or defense examinations committees. Their insights into this work inspired me immensely. I wish I could achieve that kind of intelligence some day.

I am also grateful to Dr. Laif Swanson, Dr. Kar-Ming Cheung, and Dr. Fabrizio Pollara at the Jet Propulsion Laboratory for the opportunity of working with them. I really appreciate their advice and guidance that enabled me to gain from their knowledge and experience. Thanks too to Dr. Cheung for providing me with the

code of the IBM Q-coder. This facilitated my simulation work greatly. Also I wish to thank my colleague, Hayit Greenspan, for working with me. The cooperation and discussions with her have been very helpful to me.

I would like to express special appreciation to my former colleague and good friend, Dr. Yiu-Fai, Isaac, Wong, at Lawrence Livermore Laboratory, for his great generosity in opening communicating many excellent ideas to me. Our collaboration and discussions, especially in the nonlinear multiscale techniques area, have been highly fruitful. I believe I will not have another partner like him who is so dedicated to and enthusiastic about research. I also thank him for contributing the idea of trying different nonlinear filters for this nonlinear multiscale technique.

Thanks are due to my colleagues and friends – Dr. Tsuhan Chen, Dr. Ian Galton, Dr. Petros Mouchtaris, Dr. Anand Soman, Igor Djoković, and See-May Phoong – for valuable discussions, criticism, and assistance. My special thanks also go to Cheh-Ming Liu for his tremendous skills, patience, and efforts in taking the photographs of all the images in this thesis. Also thanks go to all other friends of mine at Caltech for sharing many enjoyable moments with me and making my life so colorful at Caltech.

Finally, and most of all, I am very grateful to my parents and wife. Their care, encouragement, and material support allowed me to simply involve myself in my work without worrying about anything else. I owe them so much. Any form of gratitude is hardly enough.

Abstract

Multiscale techniques have been popular methods for image and video compression. The basic idea behind these techniques is to decompose the original signals into several components of different scales, however the scale is defined, of different sizes. One then applies appropriate encoding strategies to different components to achieve compression by taking advantage of various properties. In this thesis, we review and present several new schemes of multiscale techniques using linear and nonlinear systems. Linear techniques, which use linear filters for decomposition, have been thoroughly investigated and widely applied because of their simplicity in implementation and analysis. Herein we describe how to appropriately combine these techniques in order to process the signals more efficiently and advantageously; moreover, the picture quality of the quantized images can be improved. The subband coding technique is used as the basis for these combinations. As for the nonlinear technique, we would like to take advantage of the nonlinear features of images (such as edges) in compression to achieve compressing and enhancing of the images. Herein we design several nonlinear multiresolution systems, using various nonlinear filters, to decompose the signals in a proper form. We show that, in terms of rate-distortion performance, where mean squared error is used as the distortion criterion, these schemes are close to, or even better than, JPEG standard, whereas the encoding and, especially, decoding complexity is lower than that of JPEG. We can obtain much better image quality (in the perceptual sense), however, by applying suitable simple and fast lossless compression schemes to subimages. Simulation results are demonstrated to show the advantages and feasibility of the proposed scheme. In summary, we mention the current status and future trends of compression technologies.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1. Background

Still and moving image-compression techniques are becoming more important in contemporary digital communication and storage systems because of the large potential market. Digital laserdisc, CD-Interactive, CD-ROM, electronic camera, video-phone, video conferencing, interactive image and video, HDTV, multimedia, etc., are all important applications of the image and video compression techniques [IEEE JSAC, 1987], [IEEE JSAC, 1989], [IEEE JSAC, 1993], [IEEE CSVT, 1993], [Communications of the ACM, 1991], [Kunt, 1992]. It is believed that these techniques will play very important roles in more and more applications.

Because of the vast amount of data for image and video signals, we need a high-compression capability scheme to highly reduce the bit rate and save the transmission as well as storage bandwidth. For example, one proposal for HDTV has the following specifications: bits per pixel: 24bpp, frame size: 1408×960, and frame rate: 30/sec. The bit rate for this HDTV system, therefore, is over 1Gbits/sec just for transmitting the video signals. Of course, this is too high for normal transmission channels and inefficient for high-rate channels. We need a compression scheme to greatly reduce the bit rate.

Generally speaking, lossless compression techniques can only provide a compression ratio of about 2:1. This is certainly not enough for image and video transmission systems; therefore, we have to resort to *lossy* compression techniques. If we are

granted the flexibility of allowing distortion of the original image and video signals, then the compression ratio can be as high as several thousand to 1, depending on the resulting distortion. Even if we apply a lossy compression scheme, however, the perceptual quality of the quantized images can still be lossless; this means that one can hardly tell the difference between the quantized and original images; or, even better than that of the original ones in the medium compression-ratio region, if we properly design the compression schemes. For example, in Chapter 3, we will propose a nonlinear-filter-based, multiscale pyramid system for image compression. We demonstrate that high compression and enhancement can be achieved simultaneously.

Basically, the source encoder of a video transmission or storage system is composed of three parts: an interframe compressor, intraframe compressor, and data compactor (lossless data compressor). A block diagram is shown in Figure 1.1.1. Of course, for an image-compression system, we do not need the interframe compressor. This thesis is devoted to proposing schemes for inter- and intraframe compressors using linear and nonlinear *multiscale* techniques.

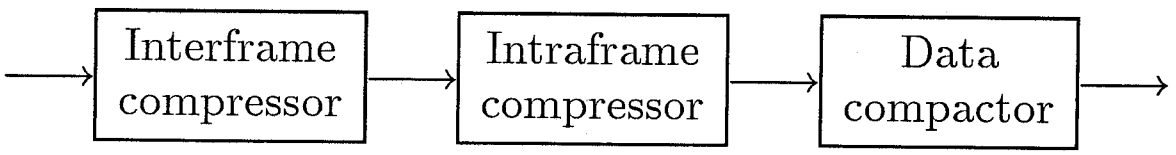


Figure 1.1.1 The source encoder of a video compression scheme.

Multiscale techniques have been popular methods for image- and video- compression systems. The basic idea behind these techniques is to decompose the original image into several components of different scales, however the scale is defined, probably with different sizes or shapes. Figure 1.1.2 shows the simple idea of these techniques. Here the signals can be a block of any shape, one frame, or several frames of images. The "black box" is to decompose the signals into several parts.

One then applies appropriate encoding strategies, inter- or intraframe, to different components to achieve compression by taking advantage of various properties.

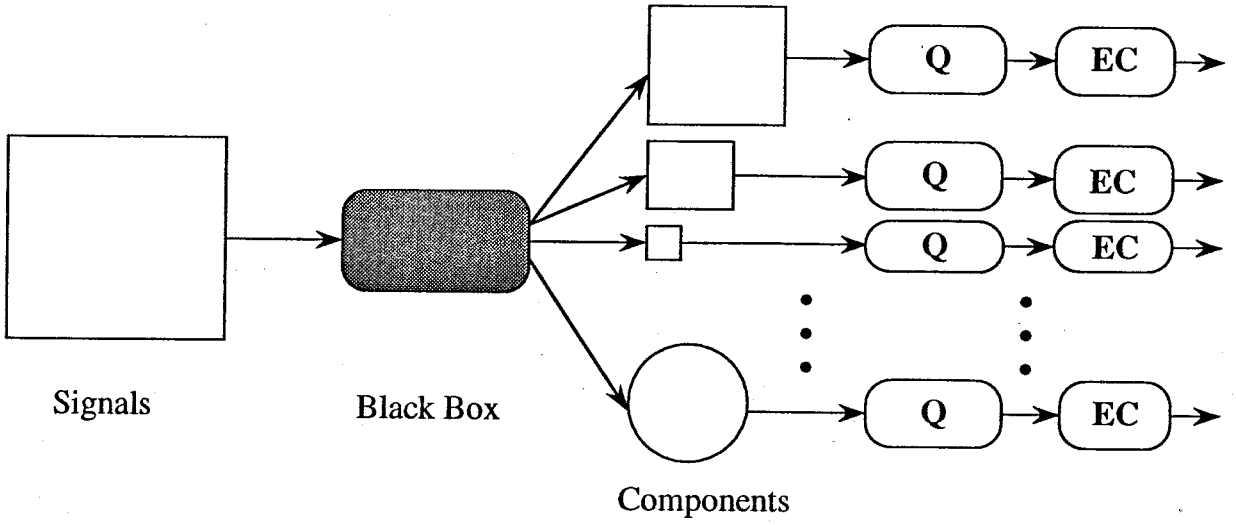


Figure 1.1.2 Basic concept of the multiscale techniques.

Linear techniques, which use linear filters for decomposition, have been thoroughly investigated and widely applied because of the simplicity in implementation and analyses. The scale defined in linear multiscale techniques is usually in a certain frequency range. The merit of these techniques is to utilize the nonuniform distribution of the signals' 2-D (images) or 3-D (videos) power spectrum and adaptively allocate bits to various components, depending on their variance, to achieve minimum distortion. Alternatively, we can first apply a motion-compensation technique to the signals to exploit the interframe redundancy, then apply a linear multiscale scheme for intraframe compression. Subband or wavelet transform-based coding, Laplacian pyramid coding, and block-transform coding all belong to this category. Most of the currently existing standards – such as JPEG [Wallace, 1991], [Pennebaker and Mitchell, 1993], MPEG [Le Gall, 1991], and H.261 [Liou, 1991] – apply this technique.

Recently, nonlinear multiscale techniques have aroused much research attention for image and video compressions because of some desired characteristics that they provide. It is well known that many important features of images are nonlinear. Furthermore, it is believed that HVS is itself a nonlinear process. In order to obtain good picture quality in the perceptual sense, therefore, it is natural to apply the nonlinear techniques. Moreover, in these nonlinear multiscale techniques, higher-order statistics (linear as well as nonlinear redundancy) may also be used to help compression.

1.2. Scope and Outline

This thesis contains the subjects described below.

Basic Compression Concept and Problems (Chapter 2)

The performance of the quantized images strongly depends on the compression schemes. It is important, therefore, to understand the criteria of judging a compression scheme. We address three important criteria: the compression capability (in terms of the compression ratio), the computational complexity of the scheme, and the distortion of the resulting images. We then discuss the distortion criterion in detail. We show that, as HVS is a nonlinear process, the PSNR criterion, which is still the most widely used objective criterion, can give only very rough indications of the image quality. We demonstrate this point by showing several interesting examples. We show that, even if two quantized images of the same original image are compared, the PSNR may serve as a bad reference if these two images are generated by different schemes. We then review and present several implementations, taking account of some of the features of HVS to improve the perceptual quality of the quantized images.

Nonlinear Multiscale Techniques for Image Compression (Chapter 3)

Because of the difficulty in obtaining a good objective distortion criterion based on the characteristics of HVS, it is not easy to control the image quality when using the rate-distortion (in MSE) approach. In Chapter 3, we propose an image-compression scheme that directly exploits the features of HVS. It is well known that edge is an important feature for HVS and that it is nonlinear. It is very helpful to enhance the perceptual quality if the quantized images have sharp edges. The basic idea behind the proposed nonlinear multiscale technique, therefore, is to retain as many edges as possible during compression, although the edge pixels are also quantized, while throwing away the points that are not perceptually important to HVS.

We demonstrate a multiresolution pyramid system based on the nonlinear filters for image compression. We emphasize applying the median filter in designing the nonlinear pyramid, then mention some other nonlinear filters that may also achieve similar performance. The original applications of the median filters are for image restoration, noise reduction, and image smoothing [Arce and Stevenson, 1987], [Gallagher and Wise, 1981], [Arce and McLoughlin, 1987]. Figure 1.2.1 shows an example of applying the median filter to image restoration. The basic idea behind this nonlinear multiscale system is to take advantage of the smoothing and edge-preserving properties of the median filters to construct subimages of different scales and sizes. Because of these characteristics, the subimages of various levels have the following desired properties: (1) The difference subimages have energy extremely localized to the edge points. (2) The low-resolution subimage has sharp edges and increased signal correlation in the smooth areas. We then apply appropriate compression algorithms to subimages. We show that, in terms of rate-distortion (MSE) performance, this scheme is close to, or even better than, JPEG standard, whereas the encoding and, especially, decoding complexity is lower than that of JPEG. We conjecture that, because higher-order statistics are used during compression, therefore, even though this is an over-sampled system, good rate-distortion performance can

Figure 1.2.1(a) The original Lenna image.

Figure 1.2.1(b) Corrupted by salt-and-pepper noise with probability 0.15.

Figure 1.2.1(c) Restored image using the 3×3 median filter.

still be obtained. Moreover, by applying suitable encoding strategies, we can obtain much better image quality, thus achieving high compression and enhancement simultaneously.

Linear Multiscale Techniques for Intraframe Compression (Chapter 4)

Subband coding [Woods, 1991], [Vaidyanathan, 1993], [Vetterli and Uz, 1992], [Vetterli, 1984], [Wickerhauser, 1992], whose block diagrams for the transmitter and receiver are shown in Figures 1.2.2(a) and 1.2.2(b), has found successful applications in image coding. We first pass the image signals through a set of linear filter bank and decimators. One then allocates various number of bits to the subbands, depending on the signal variance in that band [Soman and Vaidyanathan, 1991], [Vaidyanathan, 1993], [Jayant and Noll, 1984]:

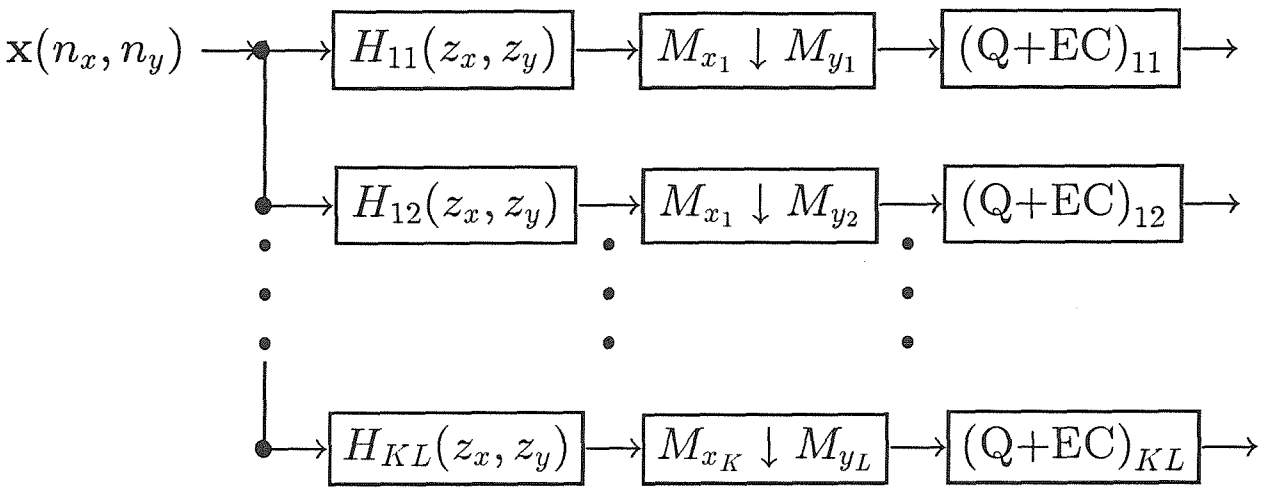


Figure 1.2.2(a) Block diagram of the subband coder in the transmitter.

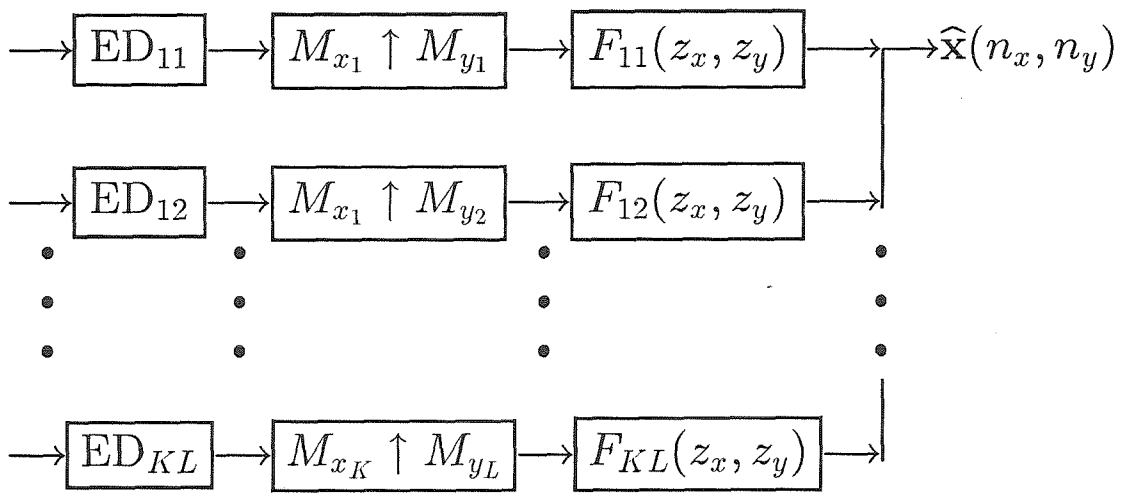


Figure 1.2.2(b) Block diagram of the subband coder in the receiver.

$$b_k = b + 0.5 \log_2 \frac{\sigma_{x_k}^2}{\left(\prod_{i=0}^{KL-1} \sigma_{x_i}^2 \right)^{1/KL}} \tag{1.2.1}$$

where

$$b = \frac{1}{KL} \sum_{i=0}^{KL-1} b_k \tag{1.2.2}$$

is a given constant, $\sigma_{x_k}^2$ is the variance of the signal in kth band, and σ_x^2 is the variance of the original signal. Because of the nonuniform distribution of the power spectrum of the image signals, one can achieve compression by allocating appropriate numbers of bits to subbands. There are several excellent reviews of this subband coding technique with applications to image compression. Compared to other linear multiscale techniques, such as DCT and Laplacian pyramid, subband coding has good compression capability. Figure 1.2.3 is an example of the comparison of the rate-distortion (in MSE) performance of DCT and 2-channel subband coding. It can be seen that the simplest 2-channel subband coding is superior to DCT in the rate-distortion performance.

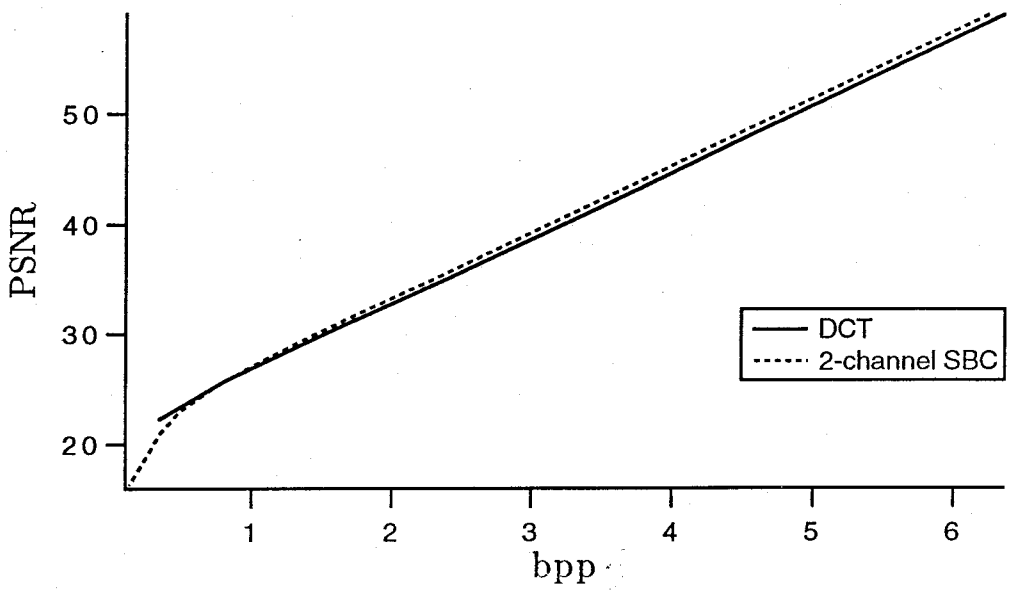


Figure 1.2.3 Rate-distortion curves of DCT and 2-channel SBC on the Baboon image.

Herein we discuss several other properties of subbanding with applications to some other coding techniques. Specifically, we consider the advantages of combining subband and block-coding systems. Basically, the subbanding process has the ability of (1) decorrelating the signals (removing the linear redundancy), (2) smoothing the block effects, and (3) transmitting the data progressively. We can apply this

technique to solve many difficulties of block-coding systems. We discuss combining with VQ and DCT. In the VQ case, we show that its encoding speed is enhanced a great deal if we apply the proposed multiscale VQ, and the quality of the quantized images is greatly improved. Recently, several proposals were made for compressing the HDTV video signals by applying the scheme of combining of subband coding and DCT. We also illustrate the advantages of this combination.

Linear Multiscale Techniques for Interframe Compression (Chapter 5)

In this chapter, we show that multirate techniques can also find applications in interframe compression [Lee, 1993b], [Lee, 1993d]. We demonstrate several interframe motion-compensation schemes for video compression. The basic idea is to use the properties of multirate techniques, including subsampling and energy compaction, to greatly reduce the complexity of the block search method. When it is properly operated, the performance of the proposed scheme can be comparable to the full-search method, whereas the complexity is much reduced. We also provide some other variant systems based on this technique to achieve different goals with higher flexibility. Simulation results are presented to show their feasibility. The MPEG system can also incorporate this technique to further improve its performance.

Linear Multiscale Techniques for Video Compression (Chapter 6)

We develop a video-compression system by combining the intra- and interframe compression schemes proposed in the previous two chapters. We take advantage of the properties of the multirate techniques in both intra- and interframe operations to greatly reduce the complexity and/or bit rate. We analyze the signal statistics in every band before and after motion compensation, then adaptively apply compression strategies. Simulation results show that high compression ratios can really be obtained, and that the performance (in terms of MSE and perceptual quality) is satisfactory. Extensions of this scheme are also mentioned.

1.3. Terminology and Abbreviations

The terminology and abbreviations used in this thesis are summarized below:

1-D: one-dimensional.

2-D: two-dimensional.

3-D: three-dimensional.

BSA: block search algorithm.

bpp: bits per pixel.

CR: compression ratio.

DCT: discrete cosine transform.

EC: entropy coder.

ED: entropy decoder.

FSA: full-search algorithm.

GMBS: generalized multiscale block search.

HVS: human visual system.

JND: just-noticeable difference.

JPEG: Joint Photographic Experts Group.

LBG: Linde-Buzo-Gray algorithm for VQ codebook generation.

MAE: mean absolute error.

MBS: multirate block search.

MPEG: Motion Pictures Experts Group.

MSE: mean squared error.

PR: perfect reconstruction.

PSNR: peak signal-to-noise ratio.

QMF: quadrature mirror filters.

SBC: subband coding.

SQ: scalar quantization.

TSS: three-step search.

VQ: vector quantization.

Chapter 2

Basic Compression Concept and Problems

2.1. Introduction

Before presenting image and video compression schemes, we first discuss the criteria of judging a compression scheme. It is important to have a thorough understanding of these because the ultimate performance of the whole system strongly depends on these criteria. Generally speaking, there are three basic and most important criteria: (a) The compression capability of the scheme. We usually use the compression ratio (CR), which is defined as

$$\text{CR} = \frac{\text{No. of original bits}}{\text{No. of compressed bits}}, \tag{2.1.1}$$

to represent it. (b) The distortion of the quantized pictures, however the distortion is defined. (c) The complexity, or the processing speed, of the scheme in both encoding and decoding. Depending on different applications, we should put different emphases on these criteria. The performance of the resulting images also strongly relies on how we put emphasis on these items. For example, for a real-time, two-way transmission system, such as video-conferencing and video-phone, both encoding and decoding complexity cannot be high to avoid too long a delay. We have to design a low-complexity system for encoding and decoding, therefore, to reduce the transmission delay and relax the compression capability and distortion

constraints; i.e., sacrifice some performance for these two. For real-time decoding systems (asymmetrical systems), however, such as storage and interactive video systems, because on-line encoding is no longer required, we can take advantage of this to build a higher compression capability scheme or provide pictures with better quality. Of course, for an off-line system, we can have an even higher flexibility for the other two criteria and obtain better performance. We feel that, at this stage, the computational complexity is typically the most important issue in designing a compression scheme for practical applications. This is why most of the current popular standards – such as JPEG [Wallace, 1991], [Pennebaker and Mitchell, 1993], MPEG [Le Gall, 1991], and H.261 [Liou, 1991] – do not have high complexity. We really need to count on the VLSI development to relax this constraint to improve the compression schemes.

Among these three criteria, the compression capability and the computational complexity are much easier to define. They can be explicitly indicated by some numbers; however, it is difficult to define a good objective distortion criterion. This is because, for most of the applications, human eyes are the ultimate viewer, and how human eyes sense the quantization noise under various conditions should be an important reference for us in designing the compression schemes. We have to take account of the properties of the human visual system (HVS), therefore, to build the distortion criterion. However, the complex behavior of HVS results in the difficulties of building a good objective criterion. This chapter is mainly devoted to discussing the distortion criterion and presenting several schemes that take account of the HVS to improve the picture quality.

Chapter outline

We first evaluate the most widely used objective criterion: MSE, or PSNR, which is defined as

$$\text{PSNR} = 10 \log_{10} \frac{255^2}{\frac{1}{XY} \sum_{i=1}^{X} \sum_{j=1}^{Y} (I_{ij} - \hat{I}_{ij})^2} \tag{2.1.2}$$

where $X \times Y$ is the frame size, I_{ij} is the original pixel value at position (i, j), $\widehat{I}_{ij}$ is the quantized pixel value, and the denominator represents MSE to show when it fails to judge pictures and when it is approximately representable. We give examples to illustrate that MSE or PSNR can in some cases only be a very rough indication of the image quality. From this, we also conclude some features of HVS. These are in Section 3.2. In Section 3.3, we review and demonstrate several schemes that take account of some of the properties of HVS to somehow enhance the picture quality. The major applications are to the block-coding systems, such as VQ and DCT (JPEG). We show that, by applying these schemes, the image quality can be slightly improved. In Section 3.4, some open problems related to this are addressed. These open issues exist mainly because the behavior of HVS is not fully understood. Finally, in Section 3.5, we give some concluding remarks about this work.

2.2. PSNR as the Distortion Criterion

By far, the most popular and widely used objective criterion is still the MSE, or PSNR, mainly because of its simplicity in analyses. We can formulate the rate-distortion performance and bound by use of the PSNR criterion. Generally speaking, it is a fairly good criterion. We can roughly tell the quality of the quantized image compared to the original one from the PSNR value. This criterion, however, cannot be totally counted on to judge pictures; otherwise, we will be very much misled. Herein we demonstrate two occasions in which the PSNR gives us a very bad indication about the picture quality.

Two image sets in the same compression scheme

The first interesting example occurs when two different image sets generated by the same compression scheme are compared. The PSNR is solely used as the criterion. Sometimes the conclusion obtained from the PSNR's indication is

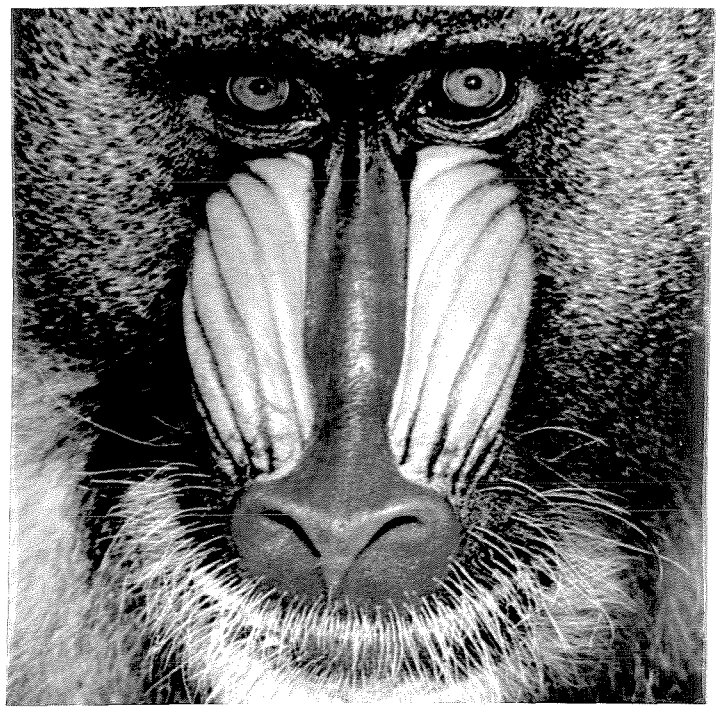


Figure 2.2.1(a) The original Baboon image.

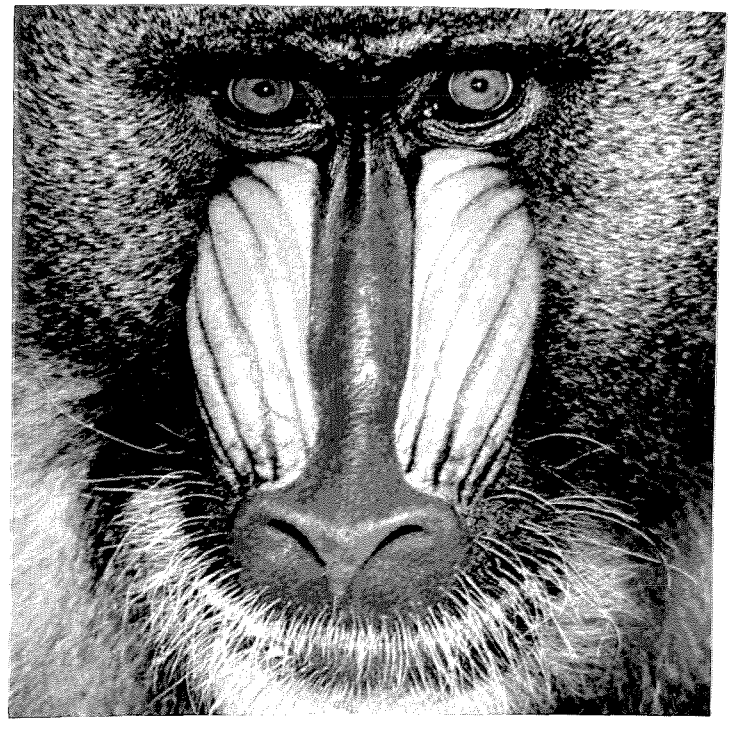


Figure 2.2.1(b) Quantized image using 2-channel SBC. The PSNR is 29.9dB.

Figure 2.2.2(a) The original Lenna image.

Figure 2.2.2(b) Quantized image using 2-channel SBC. The PSNR is 33.3dB.

totally contrary to that from HVS. Figure 2.2.1(a) is the original Baboon image. Figure 2.2.1(b) is the quantized image using 2-channel (resulting in four bands in two-dimensional case) subband coding. The PSNR is 29.9dB. Figure 2.2.2(a) is another original Lenna image (reshown of Figure 1.2.1(a) for easy comparison). Figure 2.2.2(b) is the quantized image using exactly the same compression scheme as in Figure 2.2.1. All of the parameters, including the filter coefficients and the quantization stepsize, are set to be the same in both cases. The PSNR of 2.2.2(b) is 33.3dB. We can see that, although this image has a much higher PSNR than 2.2.1(b) (3.4dB higher), perceptually it is much worse than 2.2.1(b). We can hardly tell the difference between 2.2.1(a) and 2.2.1(b), even with such a low PSNR, whereas very annoying distortion can be perceived in 2.2.2(b); e.g., in the face and shoulder areas. This demonstrates that, in this situation, the PSNR criterion may cause problems in judging the image quality.

An image set in two compression schemes

Figure 2.2.3 Quantized image of 2.2.2(a) using 4×4 VQ. The PSNR is 31.53dB.

The PSNR criterion also causes problems in judging the same image set obtained by two different compression schemes. Figure 2.2.3 is the quantized image of 2.2.2(a) using 4×4 vector quantization. The PSNR here is 31.5dB. We can see that there is another kind of distortion – namely, blocking effects – strongly visible in the quantized image. The perceptual effect seems to be better than that of 2.2.2(b), however, although the PSNR is lower by 1.8dB. This shows that, even if we compare the same image set, the PSNR criterion may also serve as a bad reference for judging pictures obtained by two different compression schemes.

From the above two examples, we see that the PSNR sometimes serves as a bad indication of the image quality. We cannot, therefore, totally rely on it to judge pictures. We suggest the most appropriate case to be that in which the same image set is compared in the same compression scheme with some parameters of the scheme being changed. Then the PSNR can be a good criterion because the distortion patterns in the same scheme are quite similar. The PSNR can now be an indication of how severe the distortion pattern is.

2.3. Compression Schemes Taking Account of HVS

Because the behavior of HVS has not been fully understood, we can only take account of part of the known properties, most of which are experimental results, for compensation to enhance the perceptual image quality. Herein we separately describe three schemes for this purpose. Combinations of these schemes are straightforward.

Intensity sensitivity

Human eyes have different intensity discrimination to noise if the values of the pixel intensity are different [Stockham, 1972], [Nasrabadi and King, 1988], [Lim, 1990], [Schalkoff, 1989]. Experiments showed that the just-noticeable-difference (JND) ΔI with respect to the intensity I behaves approximately as in Figure 2.3.1. From this figure, we see that, for a large range of I,

$$\frac{\Delta I}{I} \simeq \text{ constant.} \tag{2.3.1}$$

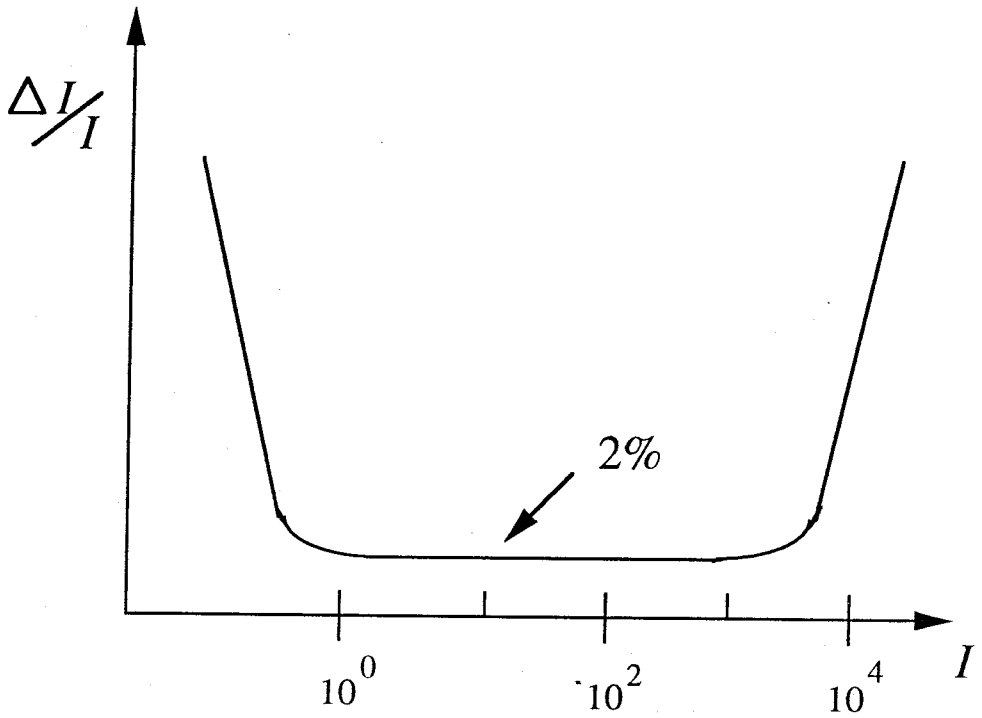


Figure 2.3.1 The JND ΔI of HVS with respect to I.

This relationship is called *Weber's law*. As I increases, therefore, we need to increase ΔI to perceive the distortion. Equation (2.3.1) can also be approximated as

$$\frac{dI}{I} = d(\log I) \simeq \text{ constant.} \tag{2.3.2}$$

So JND with respect to I is almost constant for a wide range of I. Based on this experiment, it was proposed that we should preprocess the image signals by taking the logarithm of the pixel values before the compression schemes [Stockham, 1972], [Budge et al., 1988], [Xie and Stockham, 1991], [Bradley et al., 1992], [Baseri and Mathews, 1992]. In the receiving end, we then restore them by taking the exponential. In this situation, the PSNR value obtained during the pre- and postprocessing stage, considering the above factor, should be a good indicator.

Randomness sensitivity

The sensitivity of human eyes to the noise also strongly depends on the *randomness* of the image [Lee, 1993c], [Lim, 1990], [Puri and Aravind, 1991], [Vaisey

and Gersho, 1992]. We can see this from Figures 2.2.1 and 2.2.2. In Figure 2.2.1, because the "average randomness" is very high (many texture regions), we cannot perceive the quantization noise, although it is large (the PSNR is 29.9dB, corresponding to root MSE of 8.16). In Figure 2.2.2, on the other hand, because there are many smooth regions, small quantization noise can be perceived (the PSNR is 33.3dB, corresponding to root MSE of 5.5), which results in annoying visual quality. This shows that on average, the higher the randomness of a region, the lower the sensitivity of HVS to the noise, or the higher the quantization noise allowed.

We should take account of this factor in designing a compression scheme; otherwise, the visual effect will be different in different regions. For example, in Figure 2.2.2, in the hat feather area, we cannot perceive much distortion partly because of the high randomness in that area, whereas in the face and shoulder area, large distortion can be seen. Inconsistent perceptual quality is observed, therefore, in this image. In some sense, this is inefficient in bit rate because the annoying distortion dominates the perceptual effect, so that we tend to ignore the regions with better quality. This means that we waste bits in these better-quality regions. The visual quality in every region should, therefore, be approximately the same.

We propose an *adaptive quantizer* to deal with this problem [Lee, 1993c]. The basic idea is to adaptively adjust the quantization stepsize in every region so that more quantization error is allowed in the areas where noise is less visible, whereas less error is introduced in the more sensitive areas. In other words, we adjust the quantization stepsize so that the perceptual distortion is roughly constant everywhere. Smooth or low-detailed regions, where noise is more visible, should have finer stepsize, whereas high-detailed regions, such as random texture, can have coarser stepsizes. Edge parts should have medium stepsizes. Entropy, which is defined as:

$$H = \sum_i P_i \log_2 \frac{1}{P_i}, \tag{2.3.3}$$

is a good measurement of randomness. We should adjust the quantization stepsize of a region depending on the entropy of the region. Herein we describe how to apply the proposed scheme to DCT. Applying it to other block-coding systems, such as VQ, is similar with necessary modifications.

Before applying DCT, we first decide the quantization parameter, as follows:

(1) Quantize the signals into several levels, since the HVS cannot sense all 256 pixel values. In simulations, we divide it into 64 levels. These quantized signals are just for computing the entropy.

(2) Compute the entropy of the block using equation (2.3.3).

(3) Divide the computed H_1 into several levels. For each level, assign a quantization parameter:

$$\begin{aligned} &\text{If } H_{1_i} \in R_i \Rightarrow Q_i, \\ &\text{If } H_{1_i} > H_{1_j} \Rightarrow Q_i \geq Q_j. \end{aligned} \tag{2.3.4}$$

(4) Apply DCT to this block.

(5) Multiply the basis quantization matrix – e.g., equation (2.3.5) – by the quantization parameter. This new matrix is the quantization matrix for the transformed block.

(6) Apply same entropy coding as in JPEG to the quantized signals.

(7) Entropy code the overhead for the quantization parameters of every block because they are highly redundant.

Applying this scheme, we can adaptively adjust the quantization noise of the block, depending on the randomness of this block, thus saving some amount of required bits, or improving the perceptual quality, with only a little sacrifice in complexity and overhead for every block.

Next, we show simulation results of the described scheme. Figure 2.3.2 is the quantized image using DCT. The bit rate is 0.39bpp and the PSNR is 32.93dB.

Figure 2.3.2 Quantized image using DCT without consideration of the properties of HVS. The bit rate is 0.39bpp, the PSNR is 32.93dB.

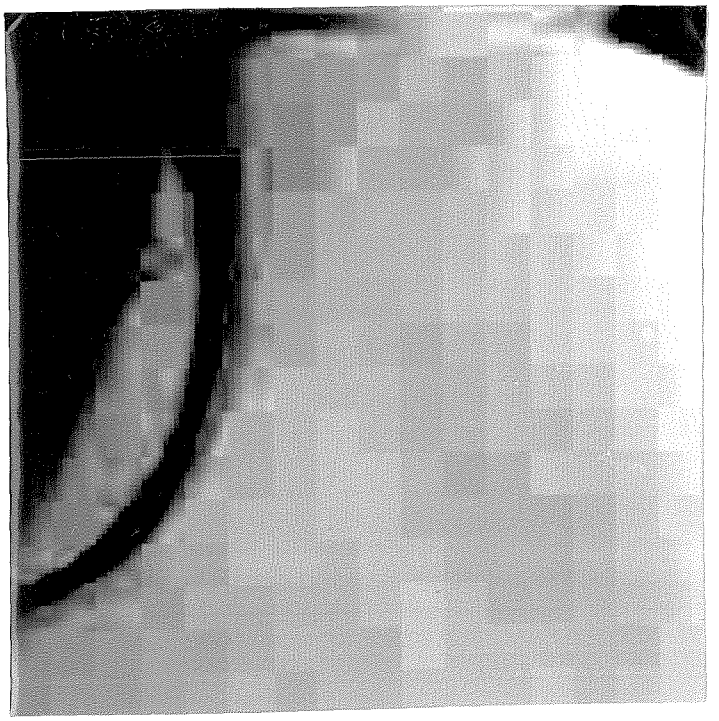


Figure 2.3.3 Quantized image using DCT taking account of randomness sensitivity of HVS. The bit rate is 0.41bpp, the PSNR is 32.47dB.

Herein we have set the quantization stepsizes of all frequency components to be the same. Figure 2.3.3 is the quantized image using the proposed scheme. The bit rate is 0.41bpp and the PSNR is 32.47dB. Comparing Figures 2.3.2 and 2.3.2, we can see that the distortion – e.g., blocking effects – is indeed less visible in the latter case, with close bit rate. This shows the feasibility of the proposed scheme.

Frequency sensitivity

Tests have also demonstrated that HVS has different sensitivity of noise in different frequency bands. We can also take account of this factor to design the quantization parameters. For example, JPEG and MPEG suggest different quantization stepsizes for different transformed coefficients after the discrete cosine transformation [Wallace, 1991], [Pennebaker and Mitchell, 1993], [Le Gall, 1991]. Each stepsize should be chosen as the approximate perceptual threshold or the JND in the corresponding frequency range. A proposed quantization matrix is as follows:

$$\mathbf{Q} = \begin{pmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{pmatrix}. \tag{2.3.5}$$

We can see that, generally speaking, HVS has higher sensitivity in the lower frequency and lower sensitivity in the higher frequency. The JND also strongly depends, however, on the spatial characteristics of the source images, displaying characteristics, and viewing distance. For example, as we have described, it is also a function of the spatial randomness of the region and the intensity of the pixels.

Some other considerations

There are certainly more factors of HVS that we can take account of to improve the picture quality or reduce the bit rate [Nasrabadi and King, 1988], [Lim,

1990], [Stockham, 1972], [Xie and Stockham, 1991]. For example, experiments have shown that HVS is much more sensitive to the luminance component than to the chrominance components [Lim, 1990], [Wallace, 1991], [Pennebaker and Mitchell, 1993], [Woods, 1991]. If we compress the color signals, therefore, we can subsample the chrominance components without noticeable performance deduction. This procedure saves large amount of bits.

2.4. Concluding Remarks

In this chapter, we have discussed some criteria of judging a compression scheme and some characteristics of HVS related to the distortion criterion. We have also presented several compression schemes taking account of these factors to improve the perceptual quality of the quantized images.

As stated, the behavior of HVS is so complex that it is almost impossible for us to use simple models to describe it. We can only apply some experimental results in designing compression schemes to slightly improve the perceptual quality of the quantized images. Just because of this, however, there is still great potential in this field. Many schemes are emerging, such as the model-based coding technique [Kunt et al., 1987], [Kunt et al., 1987], trying to directly relate their coding principles to the features of HVS. It is strongly believed that these schemes will find success and useful applications.

Chapter 3

Nonlinear Multiscale Techniques for Image Compression

3.1. Introduction

Nonlinear multiscale technique, which uses a nonlinear scale-space filter to decompose the original image into several subimages with different scales, is a relatively new technique. It recently has aroused much attention because of some desired properties [Wong and Lee, 1993a], [Wong and Lee, 1993b], [Crespo and Serra, 1993]. It is well known that linear multiscale techniques, although typically with low computational complexity, suffer from the drawback of moderate to low perceptual quality of the quantized images when the compression ratio is medium to high. For example, in transform coding, we can see strong blocking effects in the quantized images, especially when the bit rate is low or when we zoom the image up. This phenomenon results from the independent quantization of blocks. In subband coding, aliasing effects [Vaidyanathan, 1993], which are caused by the signal loss in subbands, can be seen. Part of the reason why these linear techniques cannot provide good image quality is that they do not fully take advantage of the nonlinear properties of images and HVS. We know that many features contained in images are nonlinear, and these features are important for HVS to judge the image quality because (as stated in Chapter 2) HVS itself is a nonlinear process. For example, edges [Lim, 1990], [Schalkoff, 1989], which are usually defined as the positions where abrupt intensity

change occurs, are believed to be important characteristics for both pattern recognition and HVS. Given the boundary of an object, human eyes can accurately tell what the object is without resorting to the pixel values. Linear filters tend to blur edges; therefore, exact edge locations cannot be detected in subimages during the compression process to help enhance image quality. Although the MSE may still be low, typically the perceptual picture quality is not good. Especially when the compression ratio is high, we can perceive annoying distortion in both smooth and edge parts. This will cause problems for later postprocessing when edge information is needed.

The basic idea behind this nonlinear multiscale technique is, then, to keep as many important features, such as edges, as possible during compression to preserve the image quality, while throwing away the data, such as nonimpulsive noise, which are not sensible to HVS (or even deteriorate the picture quality) to help compression. When the bit rate is high, therefore, we can not only achieve compression but also enhance images because unwanted noises, which are imposed during image formation, are smoothed out. When the bit rate is low, moreover, we can still retain certain picture quality.

In this chapter, we propose a nonlinear pyramid-based coding structure [Lee, 1993e], [Lee, 1993f] to achieve this goal. We present several nonlinear filters (with emphasis on the median filter) to decompose the images into subimages of various scales. We show that very accurate edge locations can be detected in subimages. One can then apply appropriate simple and fast encoding algorithms to achieve high compression as well as enhancement by keeping the important edge information. Furthermore, we also show that the rate-distortion (in terms of MSE) performance of the proposed scheme is close to, or sometimes even better than, that of JPEG, even though the encoding and, especially, decoding complexity is lower. This is because good signal structures are created in subimages and nonlinear statistics are used. This further strengthens the feasibility of this scheme.

Chapter outline

The organization of this chapter is as follows: In Section 3.2, we first describe the pyramid structure used in this work. The theoretical principles, characteristics, and advantages of this structure are explained. This pyramid structure is similar to the Laplacian pyramid [Burt and Adelson, 1983], except that we propose to use nonlinear filters for decomposition instead of a Gaussian filter. We show that, although this pyramid structure is an oversampled system, its compression capability can be competitive to the critically sampled systems, such as subband coding and DCT (JPEG), by properly designing the filters. In Section 3.3, we present the nonlinear filter – namely, the median filter – used in this work and for the decomposition purpose. The principles of this filter are briefly discussed. We demonstrate that this median filter has three important features: (1) preserving large-scale edges, (2) removing the impulsive noises, and (3) highly smoothing the nonimpulsive noises. We explain that these features are suitable and desirable for compression. In Section 3.4, we present the interpolation filter. We propose to use the bilinear filter for interpolation. The motivation of choosing such a simple filter is explained. In Section 3.5, we describe the important features of the various subimages generated by this nonlinear pyramid. We demonstrate that, because of the properties of the nonlinear filter, the subimages of various scales have the following characteristics: (1) The difference subimages have energy extremely localized to the edge points. (2) The low-resolution subimage has sharp edges and increased signal correlation in the smooth areas. These characteristics are quite different from those generated by a linear pyramid, such as the Laplacian pyramid. In Section 3.6, we demonstrate the rate-distortion performance of the proposed scheme. Herein MSE, or PSNR, is used as the distortion criterion. We also compare it with JPEG. We show that the performance of the proposed scheme is better than that of JPEG when compression ratio is approximately greater than 8:1, the more practical region. This better rate-distortion performance is probably caused by higher-order statistics being used

and better signal structures being created in subimages. In Section 3.7, we propose quantization and lossless compression strategies to actually encode the subimages. The merit of these lossless schemes is to take advantage of the well-created structures of subimages to efficiently compress the data after they are quantized. We show that these are all simple and fast algorithms, yet high compression as well as enhancement can be achieved simultaneously. In Section 3.8, simulation results are demonstrated to support the feasibility of the proposed scheme. We also explain that the proposed scheme is very suitable for compressing the images imposed by random noises, especially those such as salt-and-pepper noises. For example, if the image-recording equipment is not good, then the noises tend to be imposed on the images. We illustrate this by showing several examples of compressing noisy images. We see that compression and restoration can be achieved at the same time without extra pre- or postprocessing. We also show results of compressing these images, using subband coding, and demonstrate the superiority of the proposed scheme in this case. Finally, in Section 3.9, discussions and concluding remarks about this work are given. Some possible extensions and future works are also mentioned.

3.2. A Pyramid Structure

The pyramid structure we use to generate subimages of various scales is shown in Figure 3.2.1, where $\boxed{M_{x_k} \downarrow M_{y_l}}$ denotes subsampling by M_{x_k} and M_{y_l} in x and y directions, respectively. Low-resolution subimages I_k are created by passing I_{k-1} through usually a low-passed filter H and the decimation box. This structure is similar to the Laplacian pyramid [Burt and Adelson, 1983]. As shown in Figure 3.2.1, in the encoder, we transmit subimages $\{D_1, D_2, \cdots, D_L, I_L\}$ obtained by

$$\begin{aligned} D_1 &= I - I_{1i}, \\ D_2 &= I_1 - I_{2i}, \\ &\cdots \\ D_L &= I_{L-1} - I_{Li}, \\ I_L, & \end{aligned} \tag{3.2.1}$$

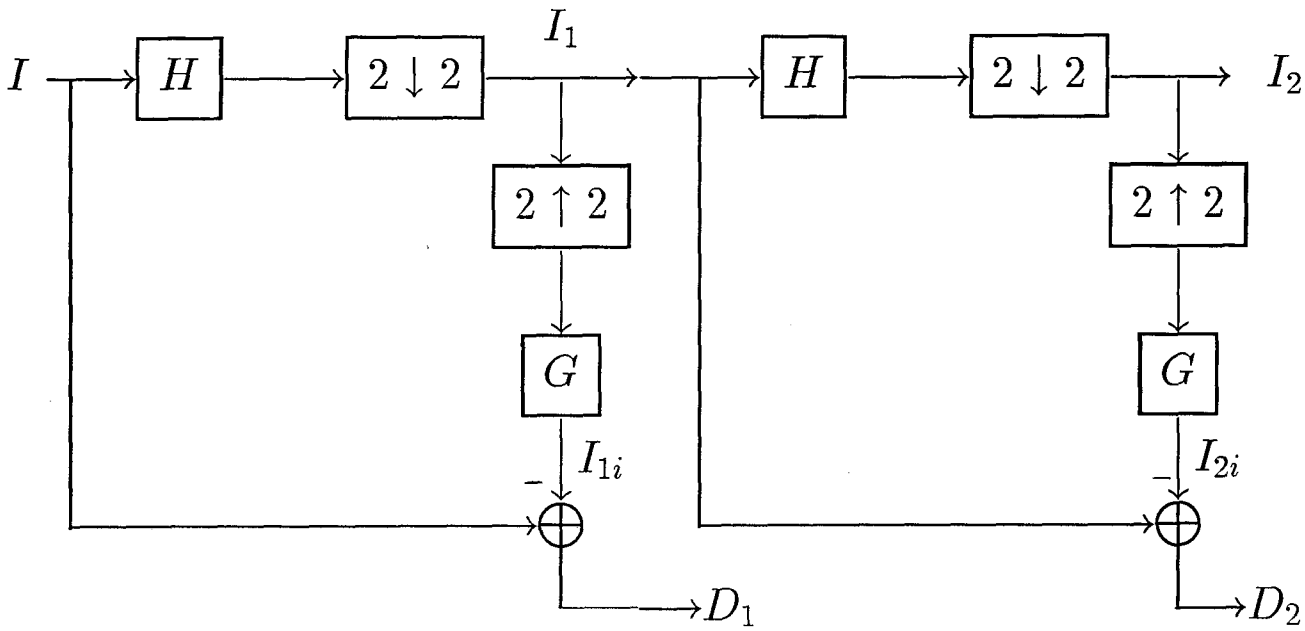


Figure 3.2.1 The pyramid structure in the encoding side.

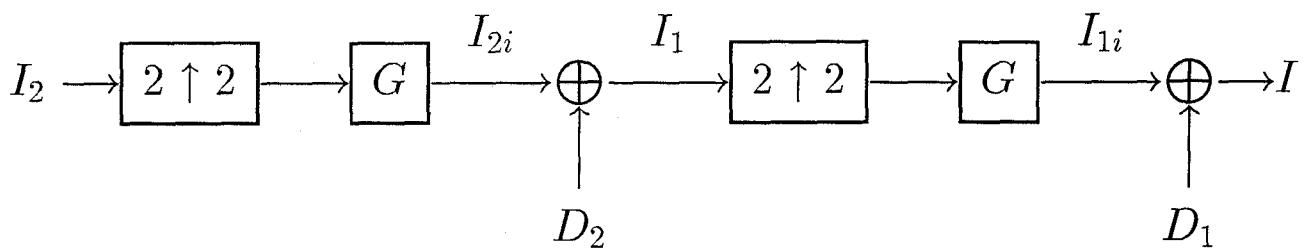


Figure 3.2.2 The pyramid structure in the decoding side.

where D_k is the difference subimage of the kth level, I_k is the low-resolution subimage of the kth level, and I_{ki} is the interpolated version of I_k. In the decoding part, we reverse equation (3.2.1) to get the original signal I. Figure 3.2.2 shows the decoding structure. Perfect reconstruction (PR) is, therefore, guaranteed if there is no quantization or transmission loss, no matter how we design the filters H and G and the decimation boxes.

Some features of this pyramid structure are described below [Burt and Adelson, 1983], [Vetterli and Uz, 1992], [Vaidyanathan, 1993]:

(a) This is an oversampled system. This means that the number of output pixels in the transmitter is greater than that of the original image. Specifically, if the original size is $N \times N$ and the decimation box is $\boxed{2 \downarrow 2}$ in every level, then the total number of output pixels is

$$\mathcal{N} = \sum_{i=0}^{L} \frac{N}{2^i} \times \frac{N}{2^i}. \tag{3.2.2}$$

Compared to subband and transform coding, which are critically sampled systems, it seems that the compression ratios we can obtain in using this structure are lower because we have to transmit more data. Because of property (b), however, this is not always true.

(b) As stated, no matter how we design H and G, we can always obtain PR; therefore, we can incorporate good and desired filters for H and G so that the signal characteristics in every level are better suited for compression and other purposes. This is a great flexibility. For example, as proposed in this chapter, we can apply a nonlinear filter for image compression or include a motion-compensated filter for video compression. This implies that we can take advantage of some nonlinear characteristics of images to help compression. Comparing to subband and transform coding, which have many constraints on designing PR systems, we can really gain many advantages here. Even though this is an oversampled system, high compression can still be achieved by properly designing H and G.

The key point here is, then, how to design good H and G filters so as to obtain good performance. We propose to use the median filter for H and the bilinear filter for G.

3.3. The Median Filter

The output of median filtering a set of data $\mathbf{x}_n = \{x_i | i = 1, \cdots, n\}$ is given by:

$$y = \text{med}(\mathbf{x}_n) = \begin{cases} x_{(v+1)}, & \text{if } n = 2v+1, \\ \frac{1}{2}(x_{(v)} + x_{(v+1)}), & \text{if } n = 2v, \end{cases} \tag{3.3.1}$$

Figure 3.3.1(a) The output image of passing the Lenna image through the 3×3 nonseparable median filter.

Figure 3.3.1(b) The difference image of the original one and Figure 3.3.1(a).

Figure 3.3.2(a) The output image of passing the Lenna image through the 5×5 nonseparable median filter.

Figure 3.3.2(b) The difference image of the original one and Figure 3.3.2(a).

where $x_{(v)}$ denotes the vth order number after sorting $\mathbf{x}_n$. For reviews of the details, properties, and implementations of the median filters, see [Gallagher and Wise, 1981], [Fitch et al., 1984], [Arce and Stevenson, 1987], [Arce and McLoughlin, 1987], and [Pitas and Venetsanopoulos, 1988]. Figures 3.3.1 and 3.3.2 show several examples of the output of passing images through the nonseparable median filters with various window sizes. We can see that there are three important features in the output images: (a) Depending on the window size, large-scale edges are still preserved very well, although some of the small-scale edges are missing. (b) The impulsive noises are removed. (c) Nonimpulsive noises in the smooth areas are highly reduced. These properties are very different from those of passing through linear filters. Linear filters tend to smear the edges and cannot totally remove the impulsive noises. After median filtering, therefore, we can still detect the locations of the large-scale edges very accurately.

Because of the properties described above, the original applications of the median filters are mainly for image restoration, noise reduction, and image smoothing. Tests have been demonstrated that for such applications, separable median filters are generally better than nonseparable ones. This is because nonseparable filters result in too much smoothing, which may destroy the original texture structures. Recently, cascade, or multistage [Arce and Stevenson, 1987], median filters have been proposed. Compared to median filters with large window size, cascade median filters can provide higher signal resolution while meeting the noise-supression specifications.

Herein we demonstrate that the described three features of the median filters are also appropriate for image compression. For generating pyramid subimages with different scales, we propose to use the 3×3 *nonseparable* filter for H. The reasons for choosing such a small-sized nonseparable filter are as follows:

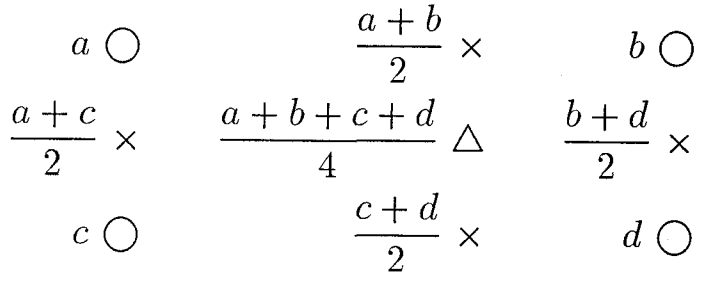
(a) If we use a larger-sized filter, then the variance of the difference subimages D_i's will be higher because more infomation is lost in low-resolution subimages.

Although we can save bits for the low-resolution subimage, we need to spend more bits for D_i's, which is not worthwhile because of their much larger image size. We would like to pass as much information as possible to the low-resolution subimage.

(b) Because we would like to take advantage of the signal correlation to achieve compression, we apply the nonseparable filter. It is expected that one pixel can be best estimated by its surrounding neighborhood.

3.4. The Interpolation Filter

From Figures 3.2.1 and 3.2.2, we know that the interpolation filter is needed on both encoding and decoding sides. Its complexity, therefore, affects both encoding and decoding speed. Furthermore, because the quantization noise must be passed through the interpolation filters, how the interpolation filters affect the final resulting quantization loss needs to be considered. We know that it is hard to describe the noise behavior after passing the quantized signal through a nonlinear filter. Taking account of these factors, we intentionally design a simple linear filter for interpolation. We apply the bilinear filter here. Figure 3.4.1 shows its operation. It can be seen that this is basically a multiplication-free process. Only simple additions



○: The received data.

×, △: The interpolated data.

Figure 3.4.1 The bilinear interpolation operation.

and shifts are needed. Thus, this part does not increase encoding complexity much, and the decoding speed is extremely fast. This is very desirable for real-time decoding systems such as storage and interactive video systems.

Although we apply such a simple bilinear filter for interpolation, because of the desired properties provided by the median filter, we can still obtain good performance in both rate-distortion (MSE) and visual-quality senses. We will demonstrate this point in a later section.

3.5. The Nonlinear Pyramid

As mentioned in the previous sections, we put the 3×3 nonseparable median filter in the H box and the bilinear filter in the G box. The low-resolution and difference subimages of various scales of the Lenna image are shown in Figures 3.5.1 and 3.5.2, respectively. Because of the properties of the median filter and the bilinear filter, these subimages have the following characteristics:

(a) The variance of the difference subimages D_i's is small (19.50, 39.51, and 80.52 in Figure 3.5.2), and the energy is extremely localized to the edge points. Other data have very small amplitude and look like random noises. We can, therefore, detect the edge information of various scales from these subimages. This is very helpful for compressing and enhancing the image quality. We will explain this later.

(b) The low-resolution subimages still have edges preserved very well. Owing to the reduction of nonimpulsive noises and removal of the impulsive noises in the smooth areas, the signal correlation among pixels is expected to be higher than the original one. This increased correlation benefits compression greatly. We can apply differential PCM (DPCM) [Petr, 1982], [Raulin et al., 1982], [Jayant and Noll, 1984] to highly reduce the bit rate. Here, in order to compare with JPEG, we apply the same DPCM scheme that JPEG does for the DC components

Figure 3.5.1 Several low-resolution subimages generated by the nonlinear pyramid.

Figure 3.5.2 Difference subimages generated by the nonlinear pyramid.

[Wallace, 1991], [Pennebaker and Mitchell, 1993]. We quantize the signals first, then take the difference between the consecutive two pixels.

Note that the complexity of the whole system is low. In the encoding part, most of the computations are spent for the median filtering, since the interpolation filter is extremely simple. Because only a small 3×3 median filter is applied, however, the complexity of this part is still low. Only comparisons are needed – a multiplication-free process. (Usually, multiplications need the most processing time compared to additions, shifts, and comparisons.) Moreover, many fast algorithms [Ahmad and Sundararajan, 1987], [Huang et al., 1979], [Rao and Rao, 1986] for median filtering have previously been proposed to speed this process. Basically, then, high-speed encoding can be achieved. As for the decoding, as stated, it is extremely fast.

3.6. Rate-Distortion Performance on Real Images

Next we investigate the rate-distortion performance of the proposed scheme. We decompose the original image I into $\{D_1, D_2, I_2\}$. We scalar quantize D_1 and D_2, then compute the entropy of the quantized signals. For I_2, we first apply DPCM as described above, then compute the entropy. The average entropy of D_1, D_2, and I_2 represents the rate. Here we use the PSNR as the distortion criterion.

Figure 3.6.1 shows the rate-distortion curves of the proposed scheme and DCT. We have tested on several real images. They all have similar rate-distortion behavior. It can be seen that, in the low-compression (high-bit-rate) region, the performance of the proposed scheme is close to that of DCT. In the medium and high-compression regions (bit rate lower than 1bpp and PSNR lower than 36dB), it is better than DCT. The distortion degradation of this scheme, when the rate is decreased, is much more graceful than that of DCT. Generally speaking, these medium and high-compression regions are the more practical ones because, in one sense, if the

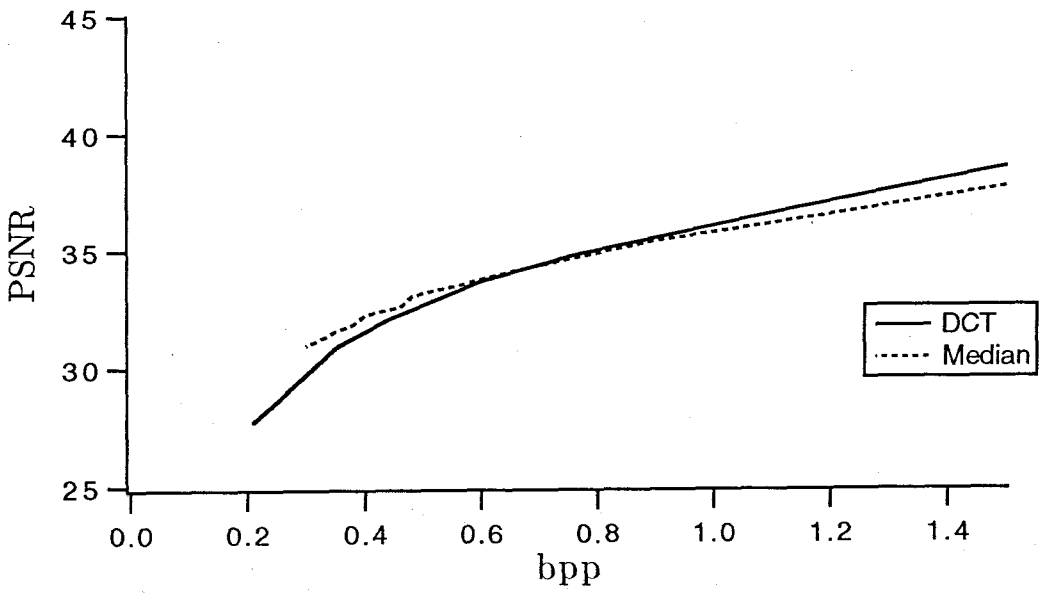


Figure 3.6.1 The rate-distortion curves of applying DCT and the nonlinear pyramid.

PSNR of the quantized image can be as high as 34dB, then typically the picture quality is expected to be good enough, although this strongly relies on the information content of the images. In the other sense, 10:1 compression is usually a minimum required ratio for a lossy image-compression scheme. Taking account of these two facts, we can say that, for practical purposes, the proposed scheme, despite its lower complexity, performs even better than DCT even in the rate-distortion sense.

The above behavior is explained as follows:

(a) In the low-compression region, because the quantization stepsize is small, those noise like signals in the difference subimages will not be zero even though their average signal amplitude is small. These data cost a certain amount of bits. Because this is an oversampled system, the total number of transmitted pixels is much greater than that of DCT. (In this case, it is 344,064 compared to 262,144 in DCT.) Therefore, we require higher rate to transmit the data.

(b) Starting from the medium-compression region, most of the noise like signals in D_i's are set to zero because of the large quantization stepsize. Because of the localization of the edge points, we do not need to spend many bits for edges. Even though we pass much information to the low-resolution subimage, however, owing to the high correlation in this subimage, applying a simple DPCM reduces a large amount of variance, thus saving many bits.

This behavior also explains the advantages of this scheme over the tranditional Laplacian pyramid. In the traditional Laplacian pyramid, even when the quantization stepsize is large (the rate is low), because the edges are blurred and wide in D_i's, we still need to spend many bits for these difference subimages. This results in a much higher bit rate because of the frame size of the difference subimages being much larger than that of the low-resolution one.

3.7. Compression Strategies

Besides the superiority of the proposed scheme in the rate-distortion sense, because of the good data structures created in the subimages, we can apply appropriate simple strategies to different subimages to obtain good picture quality and improve compression. When the bit rate is high, compression and enhancement can be achieved simultaneously. While in the low bit-rate region, we can still keep a certain level of quality by retaining important features for HVS. Next, we describe the schemes for this purpose.

As stated, the difference images D_i's have energy localized at the edge (high-detail) locations. The signal looks like random noise and has small amplitudes in the smooth areas. We observe that the small-amplitude noise does not contribute to perceptual quality. Removing it may even improve the perceptual quality. This is very desirable because noise is unavoidable during image formation, especially when the quality of the recording camera is not good. For example, in a video-phone-based transmission system, the quality of the recorded images cannot be expected

to be good because of the imposed noise. We normally need a preprocessor and postprocessor to reduce these imposed noises to enhance the picture quality. Indeed, it will be demonstrated that, with designed encoding strategies for subimages generated by this nonlinear multirescale system, we can not only save bits in coding but also get rid of the unwanted noise to improve the image-perceptual quality, yet without sacrificing the important information. No extra effort is needed, therefore, for pre- and postprocessing.

We encode the first difference image D_1 as follows: We threshold the pixel values to remove the aforementioned noise like signals; i.e., set $d_1(n_x, n_y) = 0$ if $|d_1(n_x, n_y)| <$ threshold. With proper thresholding, texture can be preserved. Then we use an edge map $e(n_x, n_y)$ to record the positions of these nonzero pixels $d_1(n_x, n_y)$,

$$e(n_x, n_y) = \begin{cases} 0, & \text{if } d_1(n_x, n_y) = 0 \\ 1, & \text{otherwise.} \end{cases} \tag{3.7.1}$$

We further throw away the isolated edge points to get rid of the impulsive noise and set the corresponding pixels to zero:

$$e(n_x, n_y) = \begin{cases} 0, & \text{if } e(n_x, n_y) = 1 \text{ and } e(n_{x-i}, n_{y-j}) = 0, \\ & \quad -1 < i, j < 1,\ i, j \neq 0 \\ e(n_x, n_y), & \text{otherwise.} \end{cases} \tag{3.7.2}$$

Even if these parts do not correspond to the impulsive noises, experiment shows that the picture's perceptual quality is not degraded, or even improved, except in some particular texture regions. To encode the pixel values, PCM is used in which we assign C bits per pixel. Thus, the transmission of D_1 consists of two parts: We first transmit the edge map, followed by the quantized nonzero pixel values.

Because of the high spatial redundancy in the edge map (its entropy H_{e_1} is much less than 1 bit/pixel for normal images), we use *run-length coding* (see [Proc. IEEE, 1980] for a review) to encode this bi-level image. Run-length code is a

variable-length-to-variable-length mapping code that is appropriate for the lossless coding of clusters or runs of like input symbols. It can lead to significant data compression with inputs characterized by long clusters of the same symbols. One-dimensional run-length code is used here. We scan the bitmap by rows, with the scanning direction reversed after each row. In designing the run-length code, we make use of the fact that, as edges are well preserved, the average run length of the nonedge pixels is much larger than that of edge pixels; that is, runs of edge pixels are usually short. Furthermore, in order to speed up the encoding process and save the codebook memory, we have decomposed a run-length b as a sum of two terms so that runs longer than 63 are broken into concatenations of shorter runs of maximum length 63:

$$b = 64 \times m + n. \tag{3.7.3}$$

We use the *modified Huffman coding* technique to design the codebook, which is a modified version of the CCITT-recommended variable-length coding system for bi-level graphics.

One can, of course, apply more complex run-length schemes, such as the 2-D prediction run-length code [Preuss, 1975], [Weber, 1975], wherein the run-length encoding is performed on the 2-D prediction error signals instead of the original ones, and the ordering technique [Netravali and Mounts, 1980], [Yasuda, 1980] where data (or predicted error data) are rearranged to enhance the run-length statistics. We intentionally choose this simple 1-D scheme, however, because we would like to build extremely fast encoding and especially decoding systems. Because of the small entropy of this edge map, only a small fraction of bits are needed to transmit it. We do not expect significant saving with more complex and expensive schemes.

For the other difference images, we use even simpler compression schemes. We first scalar quantize (uniformly) the signals. Signals with amplitude less than the quantization stepsize are discarded. This is followed by an entropy coder to further

reduce the bit rate. In simulations, the adaptive arithmetic coding (*IBM Q-Coder*) [Witten et al., 1987], [IBM, 1988] is used. The reason why we use a different compression scheme here from that in D_1 is that some of the noiselike signals may correspond to detailed regions, such as texture. Experiment shows that throwing away too many these data may affect the picture quality by missing too many details.

For the low-resolution image I_L, the correlation among the signals in the smooth areas is increased because of noise reduction and improved smoothing offered by the median filter. Thus, DPCM can be used here. We use the same simple DPCM scheme as JPEG for the DC components. This is followed by the same IBM Q-Coder on the quantized signal. Because of the much higher variance in this subimage, we can, of course, apply more complex compression schemes, such as DCT and VQ, to reduce the bit rate. We expect to gain much by applying these schemes because of high signal correlation among data. For example, if we apply DCT, then, in the smooth areas, most of the AC components will become zero after quantization. This will generate long runs of zeros in zig-zag scanning to help the lossless compression greatly. For simplicity and because of the small frame size, however, we intentionally apply such a simple scheme as that described.

We remark that the coding schemes proposed above are all very simple and fast. Next, we show some simulation results to demonstrate that high compression ratio as well as enhancement can be achieved.

3.8. Simulation Results

Herein we only decompose the image into two levels; i.e., D_1, D_2, and I_2.

The original Lenna image has been shown in Figure 2.2.2(a). Figure 3.8.1 shows an enlarged subregion. Figure 3.8.2(a) is the quantized Lenna image corresponding to the (1.33bpp, 39.2dB) point in the rate-distortion curve of Figure 3.6.1. We did not apply the aforementioned encoding strategies to this one. Figure 3.8.2(b) is

Figure 3.8.1 An enlarged subregion of Figure 2.2.2(a).

the corresponding enlarged subregion of Figure 3.8.2(a). It can be seen that, in Figure 3.8.2(b), there are some impulsive-noise-like dots, although these dots cannot be perceived from a large scale [e.g., in Figure 3.8.2(a)]. Getting rid of these noiselike dots may even improve the visual quality, and a large amount of bits can be reduced. Figure 3.8.3(a) shows the quantized image after applying the described compression strategies. Figure 3.8.3(b) is the corresponding enlarged subpart. The bit rate is now reduced to 0.93bpp (the PSNR is 37.1dB). Comparing Figures 3.8.2(b) and 3.8.3(b), we can see that the perceptual quality is indeed improved by applying the proposed encoding schemes, although with much fewer bits. Also, comparing Figure 3.8.3(b) to the original Figure 3.8.1, we see that the quality is greatly enhanced. The edges are well preserved, making the quantized image look very sharp. The noise like signals in the other parts have been greatly smoothed, making the quality even better. Compression and enhancement are, therefore achieved simultaneously.

Figure 3.8.2(a) Quantized image of 2.2.2(a) at 1.33bpp and PSNR of 39.2dB. The proposed encoding strategies were not applied.

Figure 3.8.2(b) An enlarged subregion of 3.8.2(a).

Figure 3.8.3(a) Quantized image of 2.2.2(a) at 0.93bpp and PSNR of 37.1dB. by applying the proposed encoding strategies.

Figure 3.8.3(b) An enlarged subregion of 3.8.3(a).

We can lower the bit rate by increasing the threshold in D_1 and the quantization stepsizes. Figure 3.8.4(a) shows another quantized image of bit rate 0.85bpp and PSNR 36.92dB. Figure 3.8.4(b) is the enlarged subregion. The image still looks sharp because we have retained the large-scale edges (see the enlarged image); however, we start to lose some detailed information. For example, some details are lost in the hat part. Overall, however, the picture quality is still good at this bit rate. Figure 3.8.5 is a quantized image using DCT of similar bit rate. We can see from Figure 3.8.5(b) that, at this bit rate, the quantized images of DCT start to exhibit blocking effects, although not very visible here. The proposed scheme, therefore, starts to outperform DCT in the picture-quality sense in this bit-rate range.

We have also tested this scheme on some other images. Figure 3.8.7(a) shows the quantized image of the Peppers image. The bit rate is 0.74bpp and the PSNR is 34.0dB. Figures 3.8.7(b) and 3.8.6(b) are enlarged subregions. Similar performance and conclusions to those in the Lenna image can be seen. Figure 3.8.9 is another quantized image of Figure 3.8.8, the Oilfield image, with 1.1bpp and PSNR = 31.95dB. Because of the large number of edges in this image, we need more bits to give approximately the same perceptual quality. We feel that this is very close to the behavior of HVS because practical experience tells us that we usually use the edge content to estimate the information amount in an image. Furthermore, it is believed that there are specific edge detectors in human visual systems for low-level vision tasks. As another extreme example, Figure 3.8.10 shows the quantized image of the Baboon image in Figure 2.2.1(a). Many more edges are contained in this image. We need 1.8bpp, therefore, to get similar perceptual quality.

In the high-compression (low-bit-rate) region, this scheme is superior to DCT, not only in the rate-distortion (MSE) sense, as stated in Section 3.6, but also in the perceptual quality aspect. Figures 3.8.11 and 3.8.12 show the quantized image, using the proposed scheme and DCT, respectively, with 0.36bpp and 0.39bpp. The PSNR

Figure 3.8.4(a) Quantized image of 2.2.2(a) by applying the proposed scheme. The bit rate is 0.85bpp and the PSNR is 36.92dB.

Figure 3.8.4(b) An enlarged subregion of 3.8.4(a).

Figure 3.8.5(a) Quantized image of 2.2.2(a) by DCT with 0.85bpp and PSNR of 37.76dB.

Figure 3.8.5(b) An enlarged subregion of 3.8.5(a).

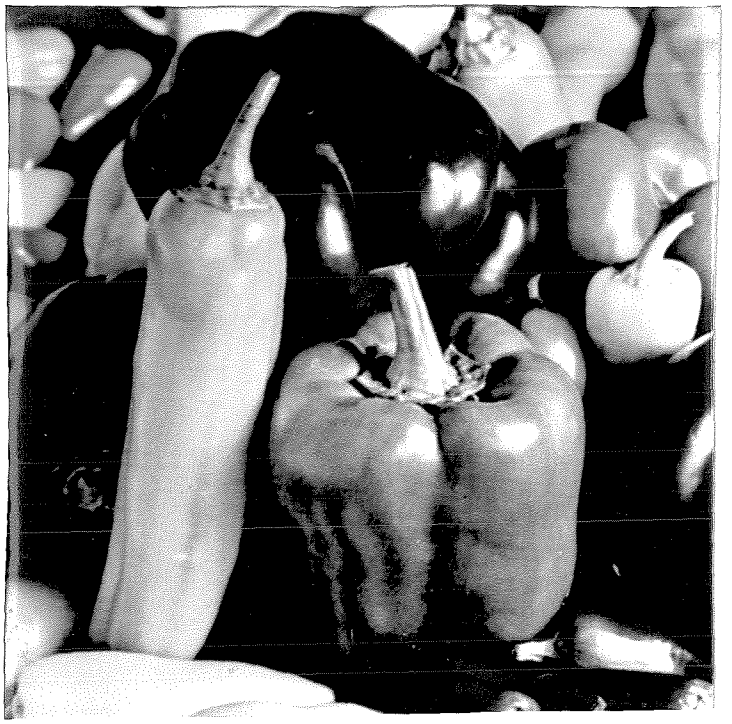


Figure 3.8.6(a) The original Peppers image.

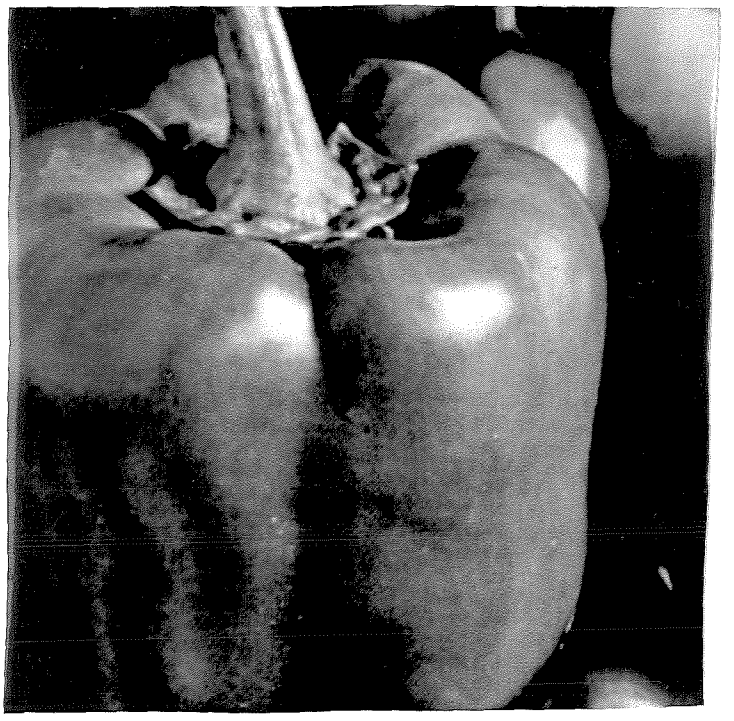


Figure 3.8.6(b) An enlarged subregion of 3.8.6(a).

Figure 3.8.7(a) Quantized image of 3.8.6(a) by applying the proposed scheme. The bit rate is 0.74bpp and the PSNR is 34.0dB.

Figure 3.8.7(b) An enlarged subregion of 3.8.7(a).

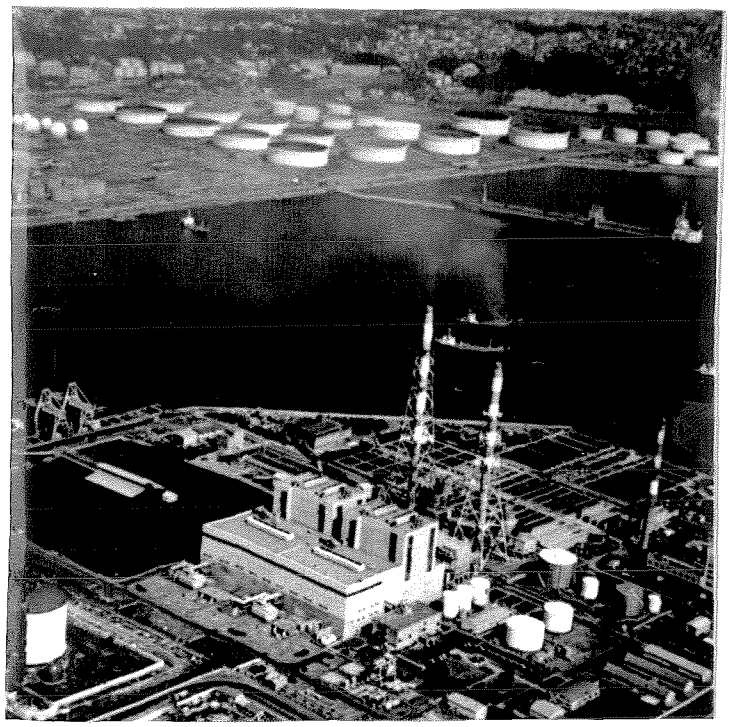


Figure 3.8.8 The original Oilfield image.

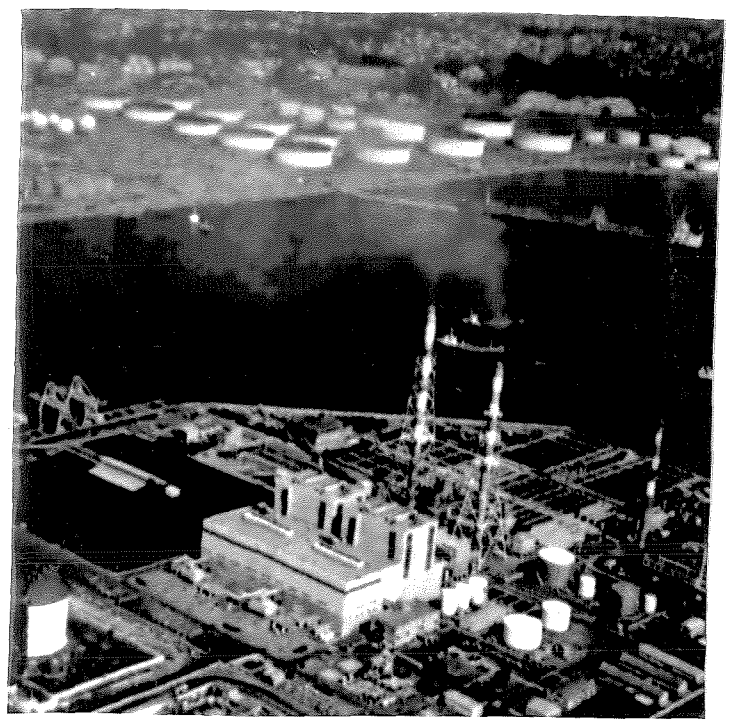


Figure 3.8.9 Quantized image by the proposed scheme with 1.1bpp.

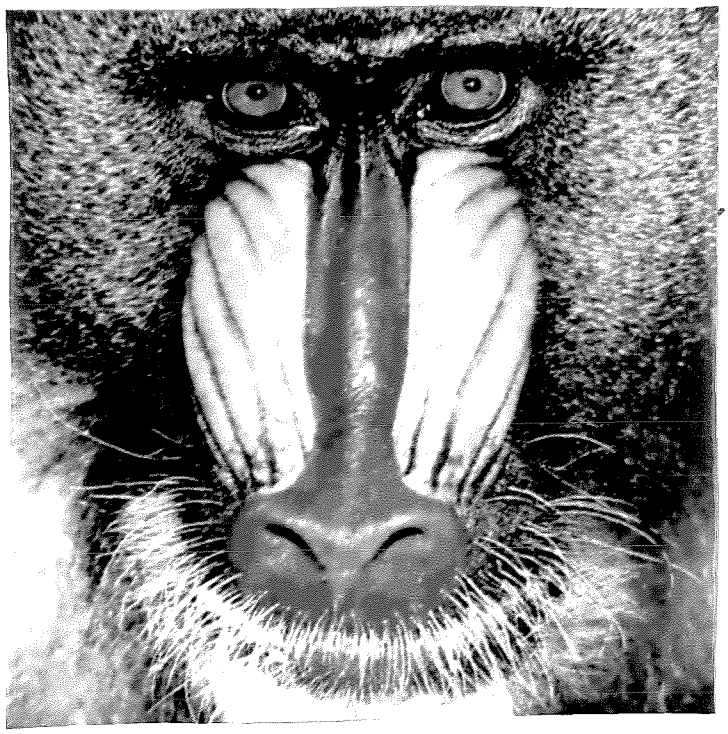


Figure 3.8.10 Quantized image of 2.2.1(a) by the proposed scheme with 1.8bpp.

values are 32.75dB and 32.92dB, respectively. We can see very strong blocking effects in Figure 3.8.12(b), a very annoying quality. Much higher quality can be perceived in Figure 3.8.11(b). This further strengthens the feasibility of the proposed scheme.

This scheme performs especially well if the transmitted images are already imposed by random noises. Compression and restoration can be achieved simultaneously. Figure 3.8.13 is the Lenna image corrupted by Gaussian random noise with mean 0 and variance 30. Figure 3.8.14 is the quantized image, using the proposed scheme with 1.07bpp and PSNR of 32.64dB compared to the original noisy image. If we compare this quantized image to the original "clean" Lenna image, however, the PSNR is 35.21dB. Restoration is indeed obtained. Comparing Figure 3.8.13(b) and Figure 3.8.14(b), an enlarged subregion, we can see that noises are greatly reduced. Other compression schemes tend to strengthen the random noise

Figure 3.8.11(a) Quantized Lenna image by the proposed scheme with 0.36bpp and PSNR 32.75dB.

Figure 3.8.11(b) An enlarged subregion of 3.8.11(a).

Figure 3.8.12(a) Quantized Lenna image by DCT with 0.37bpp and PSNR 32.03dB.

Figure 3.8.12(b) An enlarged subregion of 3.8.12(a).

Figure 3.8.13 Lenna image corrupted by Gaussian noise with zero mean and variance 30.0.

Figure 3.8.14 Quantized image of 3.8.13 by the proposed scheme with 1.07bpp and PSNR of 32.64dB and 35.32dB compared to noisy and clean original images, respectively.

Figure 3.8.15 Quantized image of 3.8.13 by SBC with 1.15bpp and PSNR of 34.50dB and 31.57dB compared to noisy and clean original images, respectively.

because of the quantization error. For example, Figure 3.8.15 is the quantized image of Figure 3.8.13, using 2-channel subband coding. The bit rate is 1.15 bpp and the PSNR is 34.50dB. Compared to the original Lenna, the PSNR is 31.57dB. Noise is increased here. Of course we can add a pre- and/or postprocessor to these schemes to reduce the noise; however, we need extra processings to achieve this.

3.9. Discussions and Conclusions

We have presented a nonlinear multiscale system based on several nonlinear filters for image compression. The motivation, structure, and advantages have been discussed in detail. Simulation results have also been shown to demonstrate the system's superiority in both rate-distortion and perceptual quality senses. This

scheme is highly suitable for low-cost and high-picture-quality transmission and storage systems, especially for real-time decoding systems.

We feel that the merit of the proposed system lies somewhere between the *first-* and *second-generation coding* [Kunt et al., 1985], [Kunt et al., 1987] techniques. First-generation coding techniques usually refer to schemes that mainly use signal statistics (spatial redundancy) to achieve compression, plus some consideration of the factors of HVS (see Chapter 4 for references). Schemes in this category include subband coding, Laplacian pyramid, transform coding, and VQ. Second-generation, or model-based, coding techniques refer to schemes that mainly use the properties of HVS first to segment the images into several components, such as edge, texture, and smooth parts, and then perform quantization separately by taking account of the sensitivity of HVS to different components. Because of the difference in motivations and goals between these two techniques, results tend to be different, especially in terms of the perceptual quality of the quantized images. At present , first-generation coding techniques still dominate the compression market, especially DCT, mainly because of its simplicity in implementation and analysis. Second-generation technique is starting to attract more and more attention, however, because it is believed that this technique matches the mechanisms of HVS more closely. Better image quality in the perceptual sense is expected be obtained, therefore, especially when the bit rate is low. There are many proposals for MPEG-4, a very low bit-rate motion video compression standard, that apply this model-based coding technique. Up to this point, the major difficulties of this scheme are its high computational complexity and not very promising compression capability. A lot of potential for this technique certainly exists.

In this proposed system, we actually have used the principles from both techniques; namely, we not only take advantage of the signal statistics to achieve compression, but also the nonlinear features of images and HVS to enhance the picture quality. We subjectively claim that edges are important features for improving the

image quality. During the compression process, we apply the nonlinear filter to extract this feature and perform high compression on the other parts. Simulation results have shown that satisfactory quality can be obtained, and the compression capability is competitive to the JPEG standard.

Certainly there are some other nonlinear filters [Pitas and Venetsanopoulos, 1988] suitable for pyramid or some other system for image compression. Depending on the properties of the nonlinear filters, the subjective performance will differ. For example, we have tested by applying the *clustering filter* [Wong, 1993b], [Wong, 1993c], [Wong and Lee, 1993a], [Wong and Lee, 1993b] to create subimages of different scales. The performance (in both MSE and visual quality) is indeed a little better than that of applying the proposed median filter (but not noticeable), partly because it can achieve even higher smoothing on the nonimpulsive noises. Because of its very high complexity in the encoding part (decoding has exactly the same structure as in the proposed system), however, we decided to apply the median filter for decomposition. Recently, the *morphological filters* [Crespo and Serra, 1993] have also been proposed for the pyramid systems.

Chapter 4

Linear Multiscale Techniques for Intraframe Compression

4.1. Introduction

Linear multiscale techniques have been popular methods for image compression. Schemes such as subband coding, DCT, and Laplacian pyramid have been proved to be powerful schemes, and are widely promoted at present. For example, DCT has been applied as the basis of many intraframe compression standards, such as JPEG, MPEG, and H.261. Thorough theoretical analyses of the various linear multiscale techniques have also been performed and reported. Excellent reviews of these techniques can be found in [Vaidyanathan, 1993], [Vetterli, 1984], [Vetterli and Uz, 1992], [Jayant and Noll, 1984], [Woods, 1991], [Wickerhauser, 1992], [Gersho and Gray, 1991], [Netravali and Limb, 1980]. Broadly speaking, the merit of these schemes is to model the images as two-dimensional random fields. One then uses the spatial correlation (linear redundancy) among the symbols or takes advantage of other structural characteristics of the signals, plus some consideration of HVS to achieve compression (e.g., schemes as stated in Chapter 2). It is well known that for general image signals, the power spectrum tends to be non uniform. We can use the *spectral flatness measure* [Jayant and Noll, 1984], which is defined as

$$\gamma_x^2 = \frac{\exp\left\{\frac{1}{2\pi}\int_{-\pi}^{\pi} \log_e S_{xx}(e^{j\omega})\, d\omega\right\}}{\frac{1}{2\pi}\int_{-\pi}^{\pi} S_{xx}(e^{j\omega})\, d\omega} = \frac{\eta_x^2}{\sigma_x^2}, \tag{4.1.1}$$

where $S_{xx}(e^{j\omega})$ is the power spectrum density of the input signals, to indicate the non flatness of the spectrum. Typically speaking, lower-frequency parts have higher energy, whereas higher-frequency parts have lower energy. Figure 4.1.1 describes the general behavior of the power spectrum of image signals. If we apply a set of filters, or transformations, to decompose the image into subimages of different scales, where the scale is defined in terms of a certain frequency range, then we can adaptively allocate a different number of bits to different components, depending on the variance of, and the sensitivity of HVS to, that band.

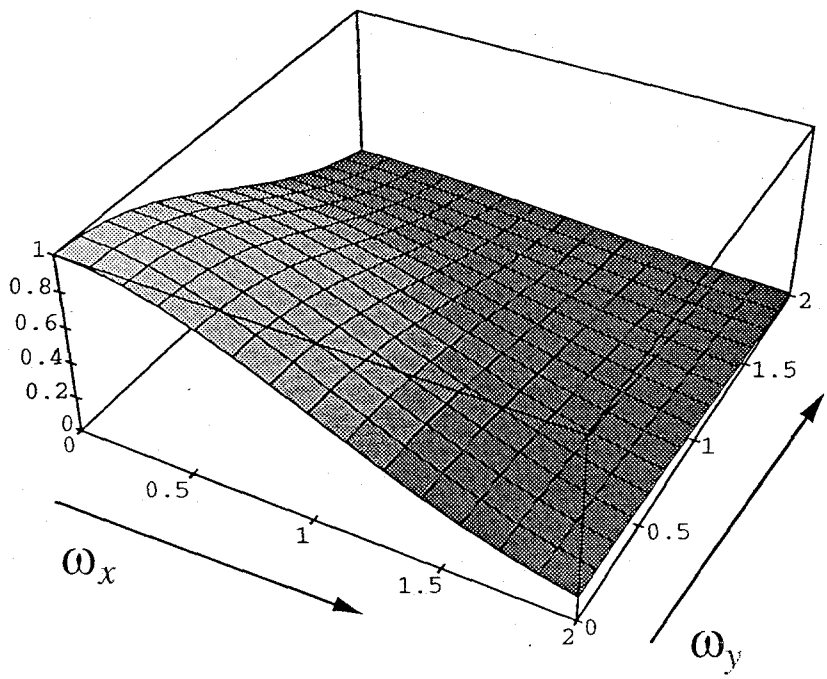


Figure 4.1.1 The behavior of the power spectrum of the general image signals.

In this chapter, we propose some other applications of the linear multiscale techniques. Herein we use the subband coding technique as the basis for the linear multiscale decomposition. We use the properties of (1) decorrelating signals and (2) progressive transmission of subband decomposition, and combine this technique with other schemes to achieve the desired purposes. For the first property, we propose the scheme of *multiscale VQ* with extremely high compression capability. The goal is to greatly alleviate the constraint of heavy encoding computational complexity of

VQ and greatly improve the quality of the quantized images. It is well known that VQ is a very powerful compression technique in the rate-distortion (MSE) sense, and has the potential of practically achieving Shannon's rate-distortion bound for any random process [Gersho and Gray, 1991], [Jayant and Noll, 1984], [Gray, 1984]. This is because VQ can exploit all orders of statistics (linear as well as non linear dependence) of the random process, whereas the other linear techniques can only remove linear redundancy.

VQ is very slow, however, in both its codebook design and encoding phases [Gersho and Gray, 1991], [Gray, 1984], [Nasrabadi and King, 1988], [Wong, 1993a], [Lee, 1993a]. How to find a good universal codebook for all kinds of images is also a major problem. Moreover, like other block-coding techniques, the quantized pictures by VQ tend to exhibit annoying blocking effects [Nasrabadi and King, 1988], [Lee, 1993a], [Wong, 1993a]. These drawbacks mean that VQ is not widely implemented in real applications, especially when real-time encoding is needed. Schemes that deal with the various aspects of VQ have been reported previously [Gersho and Gray, 1991], [Nasrabadi and King, 1988], [Foster et al., 1985], [Hang and Woods, 1985], [Kim and Lee, 1991], [Lowry et al., 1987], [Ramamurthi and Gersho, 1986], [Westerink et al., 1988], [Wu and Zhang, 1991], [Wong, 1993a], [Huang and Huang, 1992], [Ward, 1963], [Wu, 1992]. In this chapter, we propose the scheme of *multiscale PCVQ* [Lee, 1993a] to mitigate the difficulties encountered by VQ. The basic idea behind this scheme is to decompose and decorrelate the large blocks into smaller blocks, using the subbanding technique before VQ. Hence, we can apply VQ to the smaller blocks to reduce the computations greatly because the complexity of VQ is exponentially proportional to the vector dimension. This is to use the signal decorrelation property of the subband decomposition. Furthermore, in the receiving end, we have a set of synthesis filters to greatly smooth out the blocking effects, thus greatly improving the picture quality. Applying this scheme, we can comprehensively handle the aforementioned problems simultaneously.

An additional advantage of subband coding is that it is intrinsically progressive. This is achieved by dividing the original images into subimages in different bands. One then sequentially transmits subimages with (usually) increasing frequencies. Combining with the block-coding techniques, such as applying DCT in the low-low band, we can then have the advantages of both techniques. For example, we have DCT with its low-complexity property to efficiently compress the low-low subimage, but progressive transmission is still fulfilled for some other purposes; for example data browsing and image frame conversion between signal formats, such as HDTV and NTSC [Chen and Vaidyanathan, 1993], [Chen, 1993]. Although DCT can also be implemented progressively [Pennebaker and Mitchell, 1993], it is not as immediate a process. We need extra arrangement to achieve this function. Recently, there were schemes [Argenti et al., 1993], [Kim et al., 1993] proposed for compressing the HDTV signals by applying the above implementation; namely, applying DCT in the low-low band followed by an entropy encoder and SQ in the other bands followed by some other entropy encoders.

The organization of this chapter is as follows: In Section 4.2, we first address some basic properties of VQ. How it quantizes and compresses the signals and why it outperforms other coding techniques are explained. We also point out some drawbacks; namely, high complexity and low picture quality of VQ. This describes the motivations of the scheme proposed in this chapter. In Section 4.3, we briefly discuss the codebook-generation technique, the principal component algorithm, applied in this work. Extremely high speed (compared to the LBG algorithm) is its major feature, but the resulting performance is not greatly sacrificed. We explain how it works and claim that real-time codebook generation can be affordable by applying this technique. In Section 4.4, we propose the scheme of multiscale VQ. Because we use the product code VQ approach to build the scheme, we first briefly discuss the principles of product code VQ. The properties and key points of this scheme are mentioned. We then present the implementation of the proposed multiscale VQ.

How it highly reduces the complexity of VQ is discussed in detail. We will see that, because of the decorrelating property of the subbanding operation, we can save many computations for encoding signals by using VQ. We also mention some other side benefits of this scheme. We then derive the optimal codebook size allocation strategy to different bands. Here we consider only the uniform filter banks. Finally, in Section 4.5, we give some concluding remarks.

4.2. A Brief Introduction to VQ

Vector quantization (VQ) (see [Gersho and Gray, 1991], [Gray, 1984], and [Nasrabadi and King, 1988] for a review) is a relatively new data-compression technique, which has aroused much research interest during the past few years because of its potentially high compression ability and very fast decoding. According to Shannon's rate-distortion theory, we can always get better performance by coding vectors instead of scalars, even if the source symbols are memoryless.

The procedure of prototype VQ is as follows: First we define a mapping function Q of K-dimensional Euclidean space $\mathcal{R}^K$; i.e.,

$$Q : \mathcal{R}^K \longrightarrow \mathbf{Y}$$

where $\mathbf{Y} = \{\widehat{\mathbf{x}}^i | i = 1, 2, \cdots, N\}$ is the set of *reproduction vectors* of dimension K for each vector $\widehat{\mathbf{x}}^i$ and N is the number of vectors. In the encoding phase, we group a number of data to form a vector $\mathbf{x}$ of dimension K (usually taken from a square block in image coding). We then find the reproduction vector $\widehat{\mathbf{x}}^j$ that is closest to $\mathbf{x}$, commonly using the minimum MSE criterion:

$$d(\mathbf{x}, \widehat{\mathbf{x}}^i) = \sum_{k=0}^{K-1} (x_k - \widehat{x}_k^i)^2$$

$$d(\mathbf{x}, \widehat{\mathbf{x}}^j) \leq d(\mathbf{x}, \widehat{\mathbf{x}}^i), \qquad i = 1, \cdots, N. \tag{4.2.1}$$

Finally, we transmit the index j, or the entropy encoded j, of the reproduction vector. At the receiver, we get the reproduction vector $\widehat{\mathbf{x}}^j$ by searching a look up table (which is the same as the one in the transmitter) indexed by j. The block diagram is shown in Figure 4.2.1.

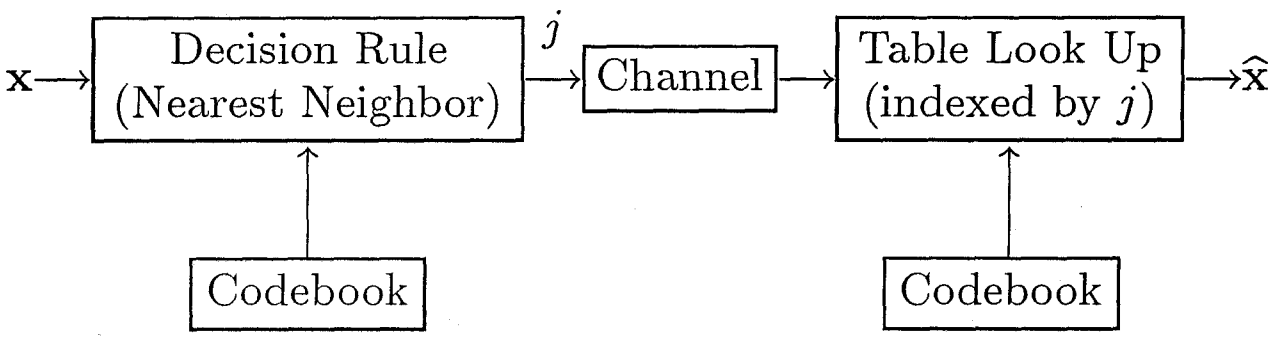


Figure 4.2.1 Block diagram of the VQ encoder and decoder.

There are some other subjective distortion measures [Gersho and Gray, 1991], such as (spectrally) weighted MSE, which will result in different performance. The choice of distortion measure, of course, depends on the criterion of judging the quantized images.

Some properties of the prototype VQ are summarized as follows:

1. Generally speaking, the performance (quantization error) of VQ strongly depends on the following:

   (i) The probability density function of the input symbols. Second-order (variance) as well as higher-order statistics all affect the resulting quality of the compressed images.

   (ii) The dependence among symbols. The stronger the dependence, the fewer bits (codevectors) are needed to get the same performance.

(iii) The number of codevectors N (or number of bits $R = \frac{1}{K}\log_2 N$). Of course, we can get better compressed-image quality by using more codevectors.

(iv) The vector dimension K. Larger vector dimension means that longer-range dependence can be included in one vector; therefore,

fewer bits are needed to get the same performance.

2. The encoding complexity per pixel of VQ is proportional to $N = 2^{RK}$; therefore, when K is large, the complexity is too high to be feasible. This is a tradeoff between performance and complexity.

3. The compressed images using VQ exhibit a severe visible blocking effect because the independent quantization of the blocks results in discontinuities at the block boundaries. The same problem also exists in other block-coding algorithms, such as DCT.

Because of the high complexity and low perceptual quality problems, VQ is still not practically feasible. Another important restricting factor is the extremely low speed of codebook design. In other words, VQ is slow not only in the encoding but also in the codebook-design phase.

In the codebook-design phase, we use the principal component algorithm proposed in [Wong, 1993a], [Huang and Huang, 1992]. Then, for the encoding phase, we propose a new scheme named *multiscale VQ*, which uses a product code VQ approach to perform compression. Combining these two algorithms, we then build a fast VQ encoding system and apply it to real-time image compression.

4.3. Principal Component Algorithm

Principal component vector quantization (PCVQ) intrinsically falls into the category of tree-structured VQ (TSVQ) [Wong, 1993a], [Huang and Huang, 1992]. Unlike the TSVQ using LBG (Linde-Buzo-Gray) algorithm [Linde et al., 1980], also

known as Generalized Lloyd Algorithm (GLA), which needs convergence operations at each stage, we continue to split the codewords until the desired number of codevectors is obtained, thus saving a tremendous amount of computations. Even though the convergence is not operated after splitting, it has been demonstrated [Wong, 1993a] that the performance is comparable to that of GLA because of the good locations of clustering centers, and the implementation speed is much higher.

Although the idea of TSVQ by splitting a codeword along the principal components of the set of vectors it encodes has been proposed before [Lowry et al., 1987], [Wu, 1992], [Huang and Huang, 1992], [Orchard and Bouman, 1991], in [Wong, 1993a], the author made it a straightforward and fast algorithm, capable of finding a good codebook much faster than any other algorithm. It can be shown that PCVQ has time complexity of $\mathcal{O}[K^2|S|\log(N)]$. Thus, PCVQ scales very well with both K and N. Simulation results show that, compared to GLA, the PSNR of PCVQ is worse only by about 0.7dB, while the speed is almost 80 times faster or more if the codebook size gets larger.

It is also noted in [Wong, 1993a] that the compressed images encoded by PCVQ tend to exhibit strong blocking effects, even more blocky than that of GLA. The author conjectured that this is because, whenever a codeword is divided into two, the cutting plane passes through the centroid. This results in the situation that nearby vectors get assigned to different codewords because the neighborhood of the centroids is more likely to be densely populated. As we will see in the next section, however, this blocking effect can be suppressed by applying the proposed "multiscale VQ" scheme [Lee, 1993a] in the encoding phase.

4.4. Multiscale VQ

In this section, we propose the scheme of multiscale VQ that can further reduce the complexity of PCVQ (TSVQ) in the encoding phase. This scheme can also greatly

reduce the annoying blocking effects. As will be seen later, the blockiness becomes almost invisible if we apply this scheme–a very great advantage.

As we use a product code VQ approach to implement this generalized multiscale PCVQ, we briefly discuss the basic idea of product code VQ first.

Introduction of product code VQ

The idea behind the product code VQ [Gersho and Gray, 1991], [Sabin and Gray, 1984] is to use a transformation to partition the vector into several subvectors with smaller dimensionality. Then we apply VQ (or SQ if the partitioned vector has dimension 1) to these smaller vectors instead of the original large vectors, thus reducing the computational complexity. Precisely, in mathematical terms, given a vector $\mathbf{X}$ of dimension $K > 1$, let $\mathbf{V}_1, \mathbf{V}_2, \cdots, \mathbf{V}_M$ be a set of vectors with dimensions lower than K that are transformations of $\mathbf{X}$. It is required that we can reconstruct $\mathbf{X}$ by using $\mathbf{V}_1, \mathbf{V}_2, \cdots, \mathbf{V}_M$. In other words,

$$\mathbf{V}_i = h_i(\mathbf{X}), \qquad i = 1, 2, \cdots, M, \tag{4.4.1}$$

and

$$\mathbf{X} = f(\mathbf{V}_1, \mathbf{V}_2, \cdots, \mathbf{V}_M). \tag{4.4.2}$$

For each $\mathbf{V}_i$, we then design a codebook $\mathbf{C}_i$ with N_i reproduction vectors from $\mathbf{V}_i$ or some training set from $\mathbf{V}_i$. A product-code vector quantizer is a quantizer that transforms the input vector $\mathbf{X}$ first and then applies VQ to each $\mathbf{V}_i$. This generates a set of indices $I_1, I_2, \cdots, I_M$ specifying the addresses of the reproduction vectors $\widehat{\mathbf{V}}_i \in \mathbf{C}_i$ for $i = 1, \cdots, M$. These are, in turn, transmitted to the decoder. At the receiving end, we have available the same codebook $\mathbf{C}_i$'s as those in the transmitter. Using these indices as well as equation (4.4.2), we can, therefore, then reconstruct the estimate $\widehat{\mathbf{X}}$ using $\widehat{\mathbf{V}}_i$ as

$$\widehat{\mathbf{X}} = f(\widehat{\mathbf{V}}_1, \widehat{\mathbf{V}}_2, \cdots, \widehat{\mathbf{V}}_M). \tag{4.4.3}$$

The block diagram of product code VQ is shown in Figure 4.4.1. Because the dimensions of $\mathbf{V}_i$'s are smaller than those of $\mathbf{X}$, we can save computations in encoding because the encoding complexity is exponentially proportional to the vector dimension.

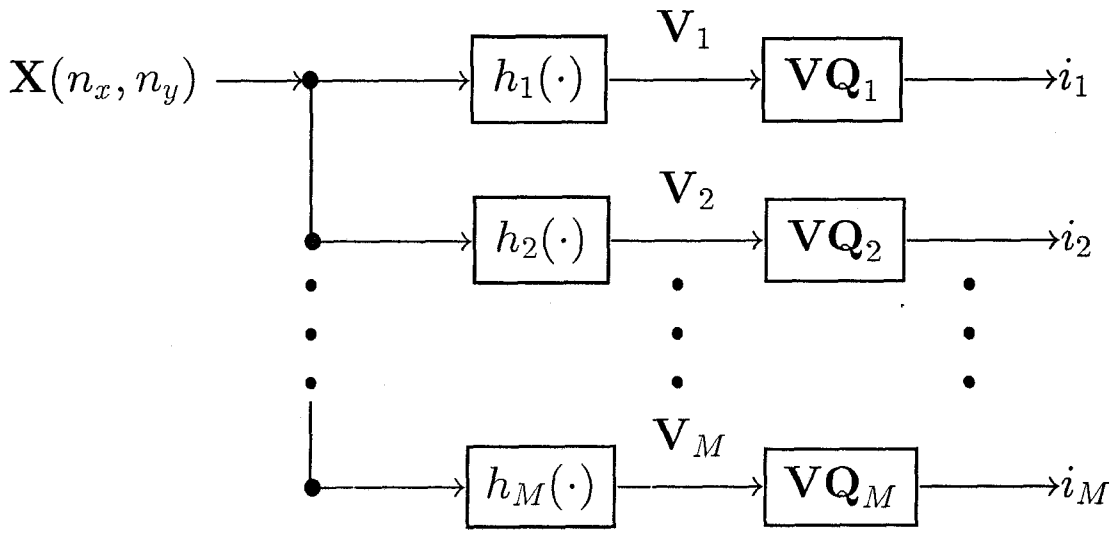


Figure 4.4.1 Block diagram of product code VQ in the transmitter.

The key point (and the goal) about the partition method equation (4.4.1) is that the functions $h_i(\cdot)$'s should have the ability to decorrelate the vector $\mathbf{X}$ so that the decomposed subvectors $\mathbf{V}_i$'s are independent of each other. Once this goal is achieved, the complexity can be greatly reduced without a significant performance reduction.

Schemes [Gersho and Gray, 1991] for product code VQ include mean-removed VQ [Baker and Gray, 1993], shape-gain VQ [Buzo et al., 1980], [Sabin and Gray, 1984], and mean/shape-gain VQ [Murkami et al., 1982]. These schemes, however, still cannot achieve the ultimate goal of product code VQ; namely, decomposing and decorrelating the vectors. The proposed "multiscale VQ" has the potential ability to achieve this.

Building block diagram of multiscale VQ

For simplicity, we consider only the case of two-dimensional *separable* filters in the following. For generality, we discuss the scheme of multiscale VQ that directly uses non uniform, maximally decimated filter banks. A block diagram of the proposed multiscale VQ is shown in Figure 4.4.2. The decimation ratios in the x and y directions are not necessarily the same; i.e., M_{x_k} may not equal M_{y_l}. This means that the support of every divided frequency band may not be square-shaped. Because the filters are separable, however, we have

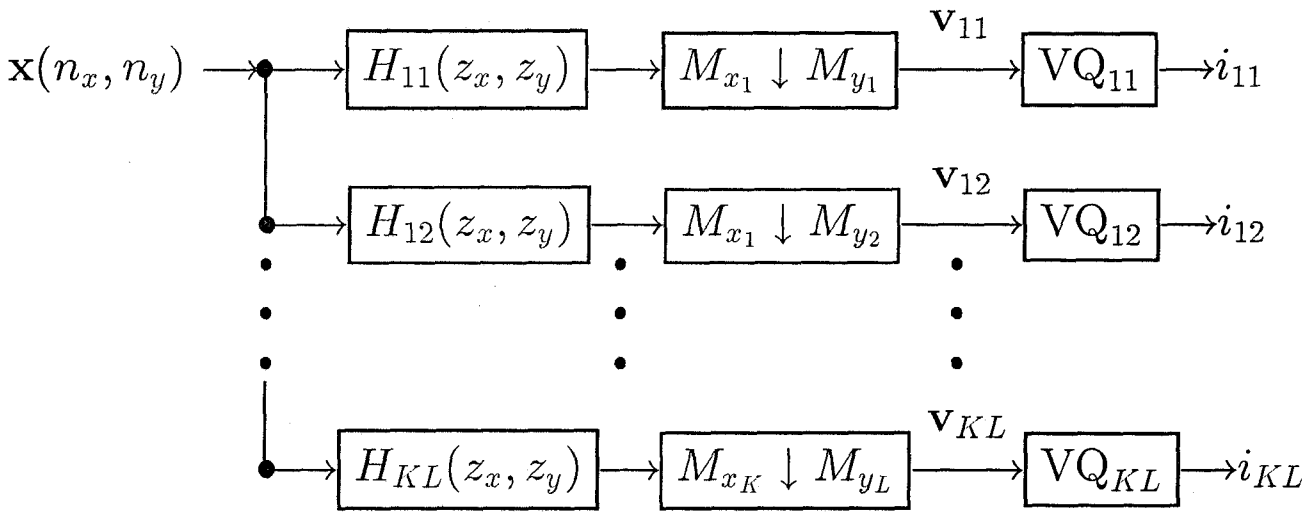


Figure 4.4.2 Block diagram of multiscale VQ in the transmitter.

$$H_{kl}(z_x, z_y) = H_k(z_x)H_l(z_y), \qquad k = 1, \cdots, K, \quad l = 1, \cdots, L, \tag{4.4.4}$$

where K and L are the numbers of channels in f_x and f_y directions, respectively (there is a total of $K \times L$ bands in two dimensions). Here the $H_{kl}(\cdot)$'s are the analysis filters. The block $\boxed{M_{x_i} \downarrow M_{y_j}}$ represents decimating (sub sampling) by M_{x_i} and M_{y_j} in x and y directions, respectively. As this is a maximally decimated system, M_{x_i}

and M_{y_j} are required to satisfy [Vaidyanathan, 1990], [Vaidyanathan, 1993]

$$\sum_{i=1}^{K} \frac{1}{M_{x_i}} = 1,$$

$$\sum_{j=1}^{L} \frac{1}{M_{y_j}} = 1. \tag{4.4.5}$$

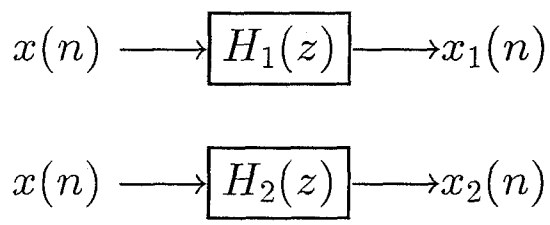
The transformed signal $v_{kl}(n_x, n_y)$ at scale k, l can be obtained by:

$$v_{kl}(n_x, n_y) = \sum_{p_y=-\infty}^{\infty} \Big(\sum_{p_x=-\infty}^{\infty} x(p_x, p_y) h_k(M_{x_k} n_x - p_x) \Big) h_l(M_{y_l} n_y - p_y),$$

$$k = 1, \cdots, K, \quad l = 1, \cdots, L. \tag{4.4.6}$$

The transformation $h_i(\cdot)$ in equation (4.4.1), therefore, corresponds to filtering and decimating in the proposed system. In order to satisfy the principle of product code VQ, we use the following well-known lemma [Vaidyanathan, 1993]:

Lemma: Assume $x(n)$ is a wide-sense stationary random process, and that $x_1(n)$ and $x_2(n)$ are the outputs of passing $x(n)$ through two filters $H_1(z)$ and $H_2(z)$, respectively:



then $x_1(n)$ is uncorrelated with $x_2(m)$ for every n and m if $H_1(z)$ and $H_2(z)$ do not overlap in the frequency domain. ♠

We can easily prove this lemma by using $\int H_1(z) H_2(z)\, dz = 0$ and deriving

$$R_{x_1 x_2}(k) = E\big[x_1(n) x_2(n+k)\big] = 0, \tag{4.4.7}$$

thus showing that x_1 and x_2 are uncorrelated.

Applying this lemma, we have the following theorem:

Theorem: An $n \times n$ block can be decomposed into several smaller blocks with size $\frac{n}{M_{x_k}} \times \frac{n}{M_{y_l}}$ in the klth band by applying the system shown in Figure 4.4.2. These sub blocks have no interblock correlation among them if the filters do not overlap in the frequency domain. In other words, the sub blocks of different scales are uncorrelated with each other if the filters are ideal. ♣

Note that the block sizes in different bands may not be the same and depend on the decimation ratios. Because of this theorem, vector quantizing these smaller $\frac{n}{M_{x_k}} \times \frac{n}{M_{y_l}}$ blocks instead of the original $n \times n$ block saves many computations, while sacrificing slightly in performance (we can only remove up to second-order interblock correlation and the filters cannot be ideally non overlapping). In other words, the resulting quantization error of applying VQ to these smaller blocks should be comparable to that on the larger blocks, and the complexity is drastically reduced. This precisely satisfies the goal of product code VQ.

The reconstructed signal is

$$\widehat{x}(n_x, n_y) = \sum_{l=1}^{L} \sum_{k=1}^{K} \Bigg(\sum_{p_y=-\infty}^{\infty} \Big(\sum_{p_x=-\infty}^{\infty} \widehat{v}_{kl}(p_x, p_y) f_k(n_x - M_{x_k} p_x) \Big) f_l(n_y - M_{y_l} p_y) \Bigg), \tag{4.4.8}$$

where $\widehat{v}_{kl}(p_x, p_y)$ is the quantized version of $v_{kl}(p_x, p_y)$ and $f_{kl}(\cdot)$ is the synthesis filter of band kl. In order to satisfy equation (4.4.2), we must choose perfect reconstruction (PR) filter banks [Vaidyanathan, 1993], [Vaidyanathan, 1990] in our scheme. Several reports on the design of the PR filter banks have appeared. For instance, we may decide to use a paraunitary (PU) filter bank [Vaidyanathan and Hoang, 1988] in simulations. Then $f_k(n)$ is related to $h_k(n)$ by

$$f_k(n) = h_k(N - n), \qquad 1 \le k \le K, L. \tag{4.4.9}$$

It is still ambiguous, however, whether a PR system can achieve smaller quantization error because there must inevitably be some loss in the subbanded domain caused by quantization. In this work, we have chosen to use PU filter banks.

One type of filter bank, called the *discrete-time wavelet transform* [Mallat, 1989a], [Mallat, 1989b], [Rioul and Vetterli, 1991] (equivalent to the tree-structured filter bank), can be easily included in our scheme without any modification. The advantage of the wavelet transform is that, it is speculated, the way it decomposes the frequency band is close to that of HVS [Mallat, 1989a], [Mallat, 1989b]. For this kind of filter bank, the block size will be smaller in the lower-frequency band and larger in the higher-frequency band.

Another benefit of our proposed scheme is that the blocking effects caused by VQ are reduced to a large extent. This is because, in the receiver end, we have synthesis filters to smooth the blockiness between the block boundaries generated in subbands. We can naturally remove the annoying blocking effects, therefore, without any further postprocessing on the compressed images. We cannot gain this advantage by using any other existing product code VQ schemes.

Optimal codebook size allocation

As stated in Section 4.1, image signals tend to exhibit different properties in different frequency bands. In particular, typical image signals have higher variance (or energy) in lower-frequency bands. If we can explore this characteristic and adaptively assign the codebook size to one band, depending on the signal variance in that band, then we can achieve more compression.

Following the above insight, we impose different bit-allocation strategies on the signals for different bands; i.e., different codebook sizes in different bands. It can be shown [Soman and Vaidyanathan, 1991], [Vaidyanathan, 1993] that, in the subband coding case (uniform SQ instead of VQ in the subbands), the optimal bit allocation

strategy is such that the variances of all quantizer noise sources are equalized; this means

$$\sigma_{q_k}^2 = \sigma_q^2, \qquad 0 \leq k \leq KL-1. \tag{4.4.10}$$

Under this condition, the number of bits allocated to the kth subband signal is ([Soman and Vaidyanathan, 1991], [Vaidyanathan, 1993])

$$b_k = b + 0.5 \log_2 \frac{\sigma_{x_k}^2}{\left(\prod_{i=0}^{KL-1} \sigma_{x_i}^2 \right)^{1/KL}}, \tag{4.4.11}$$

where

$$b = \frac{1}{KL} \sum_{i=0}^{KL-1} b_k \tag{4.4.12}$$

is a given constant, $\sigma_{x_k}^2$ is the variance of the signal in kth band, and σ_x^2 is the variance of the original signal. Equation (4.4.11) is obtained by using the formula

$$\sigma_{q,\mathrm{PCM}}^2 = c \times 2^{-2b} \sigma_x^2 \tag{4.4.13}$$

where c is a constant depending on the probability distribution of the input signal. This is a model for the quantization noise after fine scalar quantization (quantization stepsize is small).

In the VQ case, it can be shown [Gersho, 1979], [Zador, 1982], [Xie and Stockham, 1991] that, for a source distribution $p(\mathbf{x})$, the quantization noise variance of an optimal R-bit K-dimensional vector quantizer asymtotically behaves as

$$\sigma_{\mathrm{vq}}^2 = C(K,2)\, 2^{-2R}\, ||p(\mathbf{x})||_{K/(K+2)} \tag{4.4.14}$$

where $C(K,2)$ is the normalized moment of inertia of the optimal polytope for K-D uniform quantizer, a constant depending only on the dimension K, and

$$||p(\mathbf{x})||_\alpha = \left(\int p^\alpha(\mathbf{x})\, d\mathbf{x} \right)^{1/\alpha}. \tag{4.4.15}$$

This term explains the effects on the quantization error of the variances of the symbols and the dependence among the symbols in the vector. If the source $\mathbf{x}$ is Gaussian with covariance matrix $\sigma_x^2\Phi_{\mathbf{x}}$, where $\Phi_{\mathbf{x}}$ is the cross-correlation coefficient matrix and σ_x^2 is assumed to be the same for all x_i's, then

$$||p(\mathbf{x})||_{K/(K+2)} = 2\pi\sigma_x^2(\det\ \Phi_{\mathbf{x}})^{1/K}\left(1+\frac{2}{K}\right)^{1+\frac{2}{K}}. \tag{4.4.16}$$

Therefore, equation (4.4.14) can be simplified as

$$\sigma_{\text{vq}}^2 = C_1(K)\,2^{-2R}\,\sigma_x^2\,f(\rho,K). \tag{4.4.17}$$

Here

$$f(\rho,K) = (\det\ \Phi_{\mathbf{x}})^{1/K} \tag{4.4.18}$$

and

$$C_1(K) = 2\pi C(K,2)\left(1+\frac{2}{K}\right)^{1+\frac{2}{K}}, \tag{4.4.19}$$

where ρ is the normalized correlation coefficient among the symbols in one vector and $f(\rho,K)$ is a decreasing function of ρ. Substituting equation (4.4.17) into (4.4.10), we then have the following optimal bit-allocation strategy for multiscale vector quantization:

$$R_k = R + 0.5\log_2\frac{\sigma_{x_k}^2\,f(\rho_k,K)}{\left(\prod_{i=0}^{KL-1}\sigma_{x_i}^2\,f(\rho_i,K)\right)^{1/M}}, \qquad k=0,\cdots,M-1. \tag{4.4.20}$$

Note that we have assumed that the filter bank is uniform; that is, all filters have the same bandwidth. In case of a nonuniform filter bank, equation (4.4.10) has to be modified as

$$B_0\,\sigma_{q_0}^2 = B_1\,\sigma_{q_1}^2 = \cdots = B_{M-1}\,\sigma_{q_{M-1}}^2 \tag{4.4.21}$$

where B_i is the bandwidth of the ith filter. For example, in the binary tree-structured filter bank (dyadic discrete-time wavelet transform [Mallat, 1989a], [Mallat, 1989b], [Rioul and Vetterli, 1991]) case, the quantization noise variances are related by [Soman and Vaidyanathan, 1991]

$$\frac{\pi}{2}\sigma_{q_0}^2 = \frac{\pi}{4}\sigma_{q_1}^2 = \cdots = \frac{\pi}{2^{M-1}}\sigma_{q_{M-2}}^2 \tag{4.4.22}$$

$$\Rightarrow 2^{M-2}\sigma_{q_0}^2 = 2^{M-3}\sigma_{q_1}^2 = \cdots = 2\sigma_{q_{M-3}}^2 = \sigma_{q_{M-2}}^2 = \sigma_{q_{M-1}}^2. \tag{4.4.23}$$

Using this formula as the criterion of assigning the size of codebook, we can continue decomposing the low-frequency part to get a smaller vector dimension. If the vector dimension without subbanding is $n \times n$, then the dimension can be decreased to $\frac{n}{2^i} \times \frac{n}{2^i}$ at the ith stage of filtering, thus reducing the complexity of VQ substantially.

Computer simulations

To demonstrate the feasibility of the proposed multiscale VQ (PCVQ), we apply the simplest two-channel (resulting in four bands) filter banks in our simulations. We use the algorithm developed in [Vaidyanathan and Hoang, 1988] to design the filter banks. The coefficients are shown in Table 4.4.1.

n	$h_0(n)$	řule	$h_1(n)$	$f_0(n)$	$f_1(n)$
0	0.1968892	řule	−0.012578	−0.0125780	−0.1968892
1	0.5193989	řule	−0.033181075	0.033181075	0.5193989
2	0.4114735	řule	0.025083632	0.025083632	−0.4114735
3	−0.03568163	řule	0.137795	−0.137795	−0.03568163
4	−0.137795	řule	−0.035681629	−0.03568163	0.137795
5	0.025083632	řule	−0.4114735	0.4114735	0.025083632
6	0.033181075	řule	0.5193989	0.5193989	−0.033181075
7	−0.0125780	řule	−0.1968892	0.1968892	−0.0125780

Table 4.4.1 Coefficients of the two-channel analysis/synthesis filter bank.

We use five images as the training set in order to get a universal codebook. We also use the bit-allocation strategy to determine the codebook size in every subband.

Because of the difficulty of getting the variables in equation(4.4.20), becauase this formula only describes the asymtotic behavior, and because modeling the input signals to be Gaussian processes is not precise, however, we use the following method to obtain the optimum allocated codebook size: Because we want the resulting quantization noise variance in every band to be equal, we determine a final noise variance σ_q^2 in advance. According to equation (4.4.10), this σ_q^2 is also the resulting quantization noise variance for the whole system. Then, in the codebook design phase, because we apply PCVQ (a form of TSVQ), we can iterate this process and split the codevectors to increase the number of codevectors until the total quantization variance equals, or is no less than, σ_q^2. Using this method, we can obtain the optimum allocated codebook size. We use this method to design all four codebooks for the four subimages.

Figure 4.4.3 An enlarged subregion of Figure 2.2.3.

Figure 2.2.3 shows the quantized image using PCVQ with 4×4 blocks as vectors. The bit rate achieved is 0.5 bit per pixel (bpp) and the PSNR is 31.53dB. An

Figure 4.4.4(a) Quantized Lenna image using 8×8 VQ with 0.16bpp and PSNR of 31.5dB.

Figure 4.4.4(b) An enlarged subregion of 4.4.4(a).

Figure 4.4.5(a) Quantized Lenna image using multiscale VQ with 0.23bpp and PSNR of 31.53dB.

Figure 4.4.5(b) An enlarged subregion of 4.4.5(a).

Figure 4.4.6(a) Quantized Lenna image using multiscale VQ with 0.25bpp and PSNR of 32.62dB.

Figure 4.4.6(b) An enlarged subregion of 4.4.6(a).

enlarged subregion is shown in Figure 4.4.3. Figure 4.4.4 shows another quantized image using PCVQ. The vector size is 8×8, the bit rate is 0.16bpp, and the PSNR is 31.50dB. This demonstrates that a lower bit rate can be achieved if the vector dimension is larger because longer-range dependence is included, although the complexity is much greater. We can see that the PSNR in both cases is approximately the same, whereas the bit rate of the latter case is only one-third of the former one. Figures 4.4.5 and 4.4.6 show two compressed images using multiscale PCVQ. The vector sizes are both 4×4 in all subbands. The bit rates are 0.23 and 0.25bpp, respectively, and the PSNR's are 31.53 and 32.62dB.

Comparing Figures 4.4.4 and 4.4.6, we can see that the pictures compressed by applying multiscale VQ have much better perceptual quality, in spite of the fact that the complexity is reduced by a large amount. The encoding complexity per pixel is proportional to 1024 for Figure 4.4.4 and 260 for Figure 4.4.6. We can clearly see that the blocking effects in multiscale PCVQ are greatly smoothed; in fact, almost invisible. This demonstrates the effectiveness of multiscale PCVQ.

Here we have simulated only two-channel filter banks. If we were to apply a filter bank with more channels, then we could achieve a higher compression ratio, given the same performance, however that performance may be defined.

Further comments

A satisfactory way to view this multiscale VQ scheme is from Shannon's rate-distortion theory. It has been shown before that, for a wide-sense stationary Gaussian random process, we can achieve the rate-distortion bound $D(R) = \sigma_x^2 \gamma_x^2 2^{-2R}$, where γ_x^2 is the spectral flatness measure as defined in equation (4.4.1), by using VQ as the vector dimension approaches infinity; or, alternatively, using subband coding (scalar quantization in the subbands) as the channel number approaches infinity. Taking these two schemes as the two extreme architectural ends in our proposed approach, it is expected then that any intermediate compression algorithm (subbanding plus VQ with a specified block size in every frequency band) can naturally

also achieve the rate-distortion bound. We have the theoretical option, however, of minimizing the required complexity during compression as an added benefit of VQ.

In this work, we apply only separable filters. Nonseparable filters can also be used in our proposed scheme; however, the problem of how the nonseparable filter benefits image compression remains open.

Comparisons to other schemes

The scheme proposed above describes the algorithm of (generalized) *intra-subband VQ*. The idea of intra-subband VQ is not really new, and schemes of this idea have been proposed before [Antonioni et al., 1992], [Bradley et al., 1992], [Kim and Modestino, 1992]. The contribution of this work is to give a unified and theoretical approach for applying this scheme and to provide the generalized case. Unlike the previous works, which only intuitively perform VQ or other variations of VQ (motivated by and similar to the work using DPCM [Woods and O'Neil, 1986] and DCT [Argenti et al., 1993], [Kim et al., 1993]) inside every subband in order to achieve high compression ratios in that band, we use a product code VQ approach to take a different and more theoretical viewpoint of this scheme. We treat the analysis subbanding filters as the mechanism of decomposing and decorrelating larger blocks into smallers blocks so as to fulfill the goal of product code VQ. To the author's knowledge, this treatment has not been proposed before. Although the whole scheme and the results are similar, therefore,

the approach herein is totally different.

We have also been aware of the improvement in the picture quality of this scheme over that of VQ. It is possible to obtain the same compression ratio by using other constrained VQ schemes, such as predictive VQ, finite-state VQ, and adaptive address VQ, which also take advantage of the interblock correlation in coding, given the same PSNR and complexity. Blocking effects still appear, however,

in the resulting compressed pictures of the above schemes. As far as picture quality is concerned, therefore, the proposed multiscale VQ is superior to these schemes. This further strengthens its feasibility.

It should be pointed out that we can also incorporate other types of VQ, such as predictive VQ and finite-state VQ, in every subband in the proposed scheme, and thereby achieve an even higher compression ratio. We can even apply different VQ or other quantization schemes in different subbands.

There is another kind of subband VQ called *inter-subband VQ* [Westerink et al., 1988] which takes one symbol from every subband to form a vector and then quantizes these vectors. The comparisons of intra- and inter subband VQ have been made in [Bradley et al., 1992]. In this paper, it was shown that intra subband VQ is better in terms of the PSNR, given the same compression ratios.

4.5. Concluding Remarks

In this chapter, we have proposed the scheme of multiscale VQ. The motivation, principles, and simulation results have been discussed and demonstrated to show the feasibility of this scheme. Basically, the idea is to use the decorrelating property of a subbanding operation to remove the interblock linear redundancy, reducing the computational complexity and achieving compression. We have also formulated the optimal codebook size-allocation problem for this scheme. We see that the problems of high encoding complexity and low picture quality of VQ can really be alleviated by applying the proposed scheme. With this scheme, real-time encoding and decoding can be achieved.

Chapter 5

Linear Multiscale Techniques for Interframe Compression

5.1. Introduction

Interframe compression is the technique that exploits the *temporal redundancy* to further reduce the transmitted bit rate in addition to the intraframe technique. We know that typically high correlation exits among image frames. If we can appropriately take advantage of this, then we can expect to achieve high compression.

Generally speaking, there are two techniques to exploit this temporal correlation in compression. One way is, as in linear techniques for intraframe compression, to take advantage of the non uniform distribution of the signals' 3-D power spectrum and adaptively allocate bits to components, depending on the signal variance of the component. For example, we can design a 3-D subband coding system [Chen and Vaidyanathan, 1993], [Chen, 1993], [Vetterli, 1984], separable or nonseparable, or a 3-D DCT, and process the signals through this system directly to obtain compression. Basically, this is a straightforward extension from 2-D to 3-D compression. The other technique is to directly exploit the motion information among image frames to help compression. Common sense tells us that, for general video signals, almost all of the background and objects remain the same except when the scene changes. During this scene-unchanged period, the only temporal changes are mainly caused by object (or camera) motion. Consequently, nearby frames

typically exhibit high correlation. Motion-compensation techniques directly exploit this inherent temporal redundancy and significantly reduce the data rate required to transmit the video signals, thus achieving a high compression ratio. Motion estimation is the technique to approximately identify the moving objects and find the corresponding motion vectors. It is widely believed that, if the motion information can be detected accurately, this technique is much more efficient than the former one. Almost all of the video-compression standards, such as MPEG [Le Gall, 1991] and H.261 [Liou, 1991], use motion compensation to exploit the temporal redundancy.

In general, there are three main categories of motion-estimation techniques for the compression purpose; namely, pel-recursive [Biemond et al., 1987], [Moorhead II et al., 1987], contour-based [Carlsson and Reillo, 1986], [Huang and Mersereau, 1993], and block-search [Jain and Jain, 1981], [Koga et al., 1981], [Kappagantula and Rao, 1983], [Srinivasan and Rao, 1984], [Puri et al., 1987], [Puri and Aravind, 1991], [Ghanbari, 1990], [Liu and Zaccarin, 1993] algorithms. Combinations of these techniques are possible. For simplicity and easy implementation of VLSI, we usually apply the block-search technique in interframe compression. A full block-search algorithm is still too complex to be implemented, however, especially when real-time compression is needed. Several fast algorithms (see the above references) of block search have previously been proposed to speed the computation, with the tradeoff of less accurate estimation. This means that we need to spend more bits in intraframe compression.

In this chapter, we propose new schemes of *generalized multirate motion compensation* [Lee, 1993a], [Lee, 1993c] to further improve the performance of the block-search algorithm. The idea behind these proposed schemes is to use the properties of multirate techniques [Vaidyanathan, 1993], including the subsampling and energy-compaction characteristics to efficiently process the data, and the correlation of the motion vectors among adjacent pixels to reduce the computational complexity

and overhead amount. Instead of performing motion compensation on the original signals, we apply it after the multirate processing. We first decompose every frame into several subimages, using multirate technique, then appropriately apply a motion-compensation algorithm in every channel. We show that, by applying the simple subsampling and properly designing the algorithm, we can save many computations. Moreover, if we apply a filter bank first, then depend on the signals' characteristics in different bands to design various estimation strategies with different complexity, we can reduce computations even more, or obtain more accurate estimation. We can also estimate the motion vectors of some bands by using that of the others. We design several methods of performing block search in subbands, which can not only reduce the complexity but also obtain accurate estimation.

Chapter outline

The organization of this chapter is as follows: In Section 5.2, we briefly review the principles of the block-search algorithm and some fast schemes. We also point out some characteristics and drawbacks of the block-search technique, then explain that block search can give only a rough estimation of motion information. In Section 5.3, we introduce a simple multirate block search (MBS) scheme. The principles and advantages over the full-search type of this scheme are mentioned. We show that, by properly using the simple decimators (separable decimators), the complexity of block search can be greatly reduced, and accuracy can still be retained. Several extensions of this scheme are also presented. Simulation results of every proposed scheme are demonstrated to show their feasibility. In Section 5.4, we present the generalized multiscale motion-compensation scheme and some of its modified algorithms. Details of this scheme and how it reduces the complexity are discussed. We show that, by applying a filter bank before the decimators to take advantage of the non uniform distribution of the signals' power spectrum, we can obtain an even more accurate estimation. Simulation results are shown to illustrate the advantages. We also discusst some important points about the filter banks suitable

for motion estimation. Finally, in Section 5.5, we give some concluding remarks about this and discuss the possible extensions.

5.2. Review of Block-Search Algorithms

The basic idea behind the block-search algorithm (BSA) is to partition the whole image into several non overlapping regions, then estimate the velocities (motion vectors) of moving objects on a region-by-region basis without any consideration of how the objects are moving as a whole. It takes many computations to accurately detect the moving objects.

In a typical BSA, we usually divide a frame into rectangular blocks with $M \times N$ pixels or, more favorably, square blocks of N^2 pixels. Then we set a maximum displacement d between consecutive frames. If one pixel of an object is at position (i, j) in frame c, then the corresponding pixel cannot appear outside the region of $\{(x, y)|(x - i)^2 + (y - j)^2 \leq d^2\}$ in frame c. For simplicity, however, we use square search regions instead of circular search regions in BSA; i.e., the search region becomes $\{(x, y)|i - d \leq x \leq i + d, j - d \leq y \leq j + d\}$. Finally, we find the best match of every block in frame c from all the blocks in the specific search region in frame $c - 1$ (forward prediction), given a particular distortion criterion. For example, the matching (distortion) criteria, using MSE and mean absolute error (MAE), are as follows:

$$D_{s(i,j)}(m,n) = \frac{1}{N^2}\sum_{k=1}^{N}\sum_{l=1}^{N}\left(f_c(i+k,j+l) - f_{c-1}(i+m+k,j+n+l)\right)^2, \quad -d \leq m,n \leq d, \tag{5.2.1}$$

$$D_{a(i,j)}(m,n) = \frac{1}{N^2}\sum_{k=1}^{N}\sum_{l=1}^{N}\left|f_c(i+k,j+l) - f_{c-1}(i+m+k,j+n+l)\right|, \quad -d \leq m,n \leq d, \tag{5.2.2}$$

where $f_c(i,j)$ is the gray-level at pixel (i,j) of frame c. If

$$D_{(i,j)}(m_0,n_0) \leq D_{(i,j)}(m,n), \qquad -d \leq m,n \leq d, \tag{5.2.3}$$

then (m_0,n_0) is the estimated motion vector of the block at coordinate (i,j). Therefore, the prediction of $f_c(i,j)$ is

$$\widehat{f}_c(i,j) = f_{c-1}(i+m_0, j+n_0), \qquad 1 \leq i,j \leq N. \tag{5.2.4}$$

Then we apply an intraframe compression algorithm on the compensated (prediction) error signals,

$$e_c(p,q) = f_c(p,q) - \widehat{f}_c(p,q), \qquad 1 \leq p \leq X, 1 \leq q \leq Y, \tag{5.2.5}$$

where $X \times Y$ is the frame size. So, in the transmitter, we send out the overhead for motion vectors as well as the compressed error signals. If we can detect the motion vectors– the (m_0,n_0)'s, more accurately–then the energy in e_c will be less. This means that we need fewer bits in intraframe compression to get the same performance.

Several points about BSA should be mentioned herein:

(A) BSA has several drawbacks imposed by the block-search property. For example, BSA can only deal with *translational* movement. Other kinds of movement, such as rotation and zoom, fail to be accurately described by BSA. This is one major weakness of BSA. Also, occlusion of the backgrounds and objects cannot be detected.

(B) Using pure prediction, BSA cannot handle uncovered regions between frames. One solution, such as MPEG [Le Gall, 1991], besides pure prediction, is to use interpolative prediction for motion estimation; i.e., use frame c and $c+j$ to predict frame $c+i$, where $j > i$. Of course, this takes more computational complexity and overhead for motion vectors; however, it can successfully reduce

the signal variance of the compensated images by a large amount, thus greatly saving the data amount for intraframe compression–a great advantage.

(C) One moving object tends to be decomposed into different blocks. One block may also include components with different moving directions, especially when the block is around the boundaries of moving objects. This is the major reason why block search gives only rough motion information. The estimated images tend to exhibit a very strong blocking effect, therefore, and the compensated signals tend to have non uniform probability distribution. These phenomena are especially clear around the boundary parts. One way to smooth the blocking effect is to apply generalized multiscale block search, which is described in a later section.

We can see that, based on the above equations, if we apply the full search algorithm (FSA)–i.e., compare all of the possible blocks in the search area–then we need $(2d+1)^2 \times \mathcal{O}[D(\cdot,\cdot)]$ evaluations for computing the motion vector of one block, where $\mathcal{O}[D(\cdot,\cdot)]$ is the complexity needed at every position in the search area. For example, if the block size is 8×8, the maximum displacement d is 8, and the frame size is 480×640 (row $\times$ column, NTSC standard size), then we need $3 \times 64 \times 289 \times 60 \times 80 = 2.35 \times 10^8$ additions and $64 \times 289 \times 60 \times 80 = 7.83 \times 10^7$ multiplications to locate the motion vectors for the whole frame if we use MSE as the distortion criterion. (Multiplying by 30 frames/sec is the number of operations that must be done in 1 second for doing motion estimation.) This is very computationally expensive and cannot yet be carried out in real-time compression. Some fast algorithms–such as three-step search (TSS) [Koga et al., 1981], two-dimensional logarithmic search method (TDL) [Jain and Jain, 1981], modified motion estimation algorithm (MMEA) [Kappagantula and Rao, 1983], conjugate direction search (CDS) [Srinivasan and Rao, 1984], cross-search algorithm (CSA) [Ghanbari, 1990], and so on–have been proposed to speed the computations. The merit of these algorithms is that, instead of full search, they just recursively search the blocks at

certain particular coordinates, thus drastically reducing the number of positions that need to be computed. These techniques rely on the assumption of a monotonically increasing distortion around the position of the optimal motion vector to iteratively determine that position [Liu and Zaccarin, 1993], [Ghanbari, 1990], [Jain and Jain, 1981]. There are usually local minima on the distortion surface, however, so that these algorithms are trapped. These algorithms provide a tradeoff, therefore, between the complexity and the precision of the motion vectors (or necessary bit rate for intraframe compression). Figure 5.2.1 shows the principles of TSS. We can see that, if $d = 8$, then the complexity becomes proportional to 25, compared to 289 in full search, or a gain of 11.56. Its performance compared to that of full search is shown in the next section. The corresponding numbers used in applying the other schemes are: TDL: 23, MMEA: 19, CDS: 19, and CSA: 17. Recently, a new algorithm [Liu and Zaccarin, 1993] that takes advantage of the correlation of the motion vectors among blocks to save computations was proposed. It is shown later that this idea is quite close to that of one of our schemes.

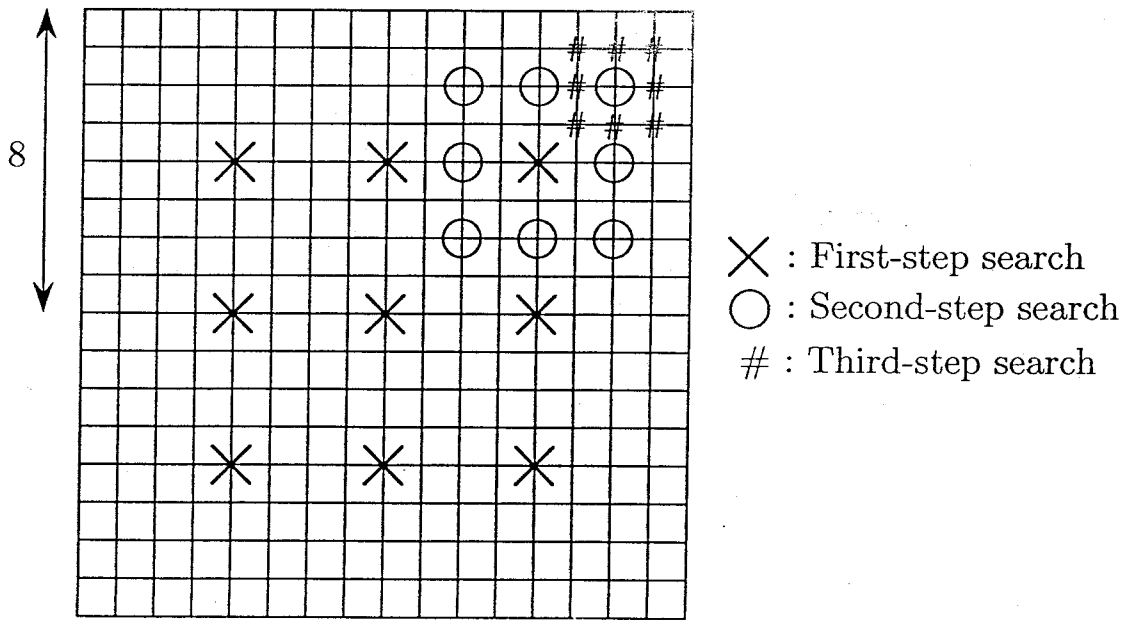


Figure 5.2.1 The procedure of TSS.

5.3. Simple Multirate Block-Search Schemes

From the above, we can see that the maximum displacement and the block size decide the precision and time complexity of motion estimation. Generally speaking, larger maximum displacement and smaller block size can get more accurate estimation, but the complexity and overhead are increased. The basic idea behind this simple multirate block search (MBS) algorithm is that we can decimate the original image signals first, then perform motion estimation on every decimated subimages with smaller maximum displacement without sacrificing accuracy. Figure 5.3.1 shows a block diagram of this scheme of 2-channel (resulting in four subimages in two dimensions) case. It is easy to show that this delay-chain is a PR system [Vaidyanathan, 1993], [Chen, 1993]. In this circumstance, the maximum displacement and the block size will also be subsampled by the corresponding numbers (e.g., 2 as in Figure 5.3.1). The complexity is now proportional to $(d+1)^2$ instead of $(2d+1)^2$ as in full search. The computational gain is then approximately 4. Because of the smaller block size in every subimage, however, more accurate estimation can be obtained.

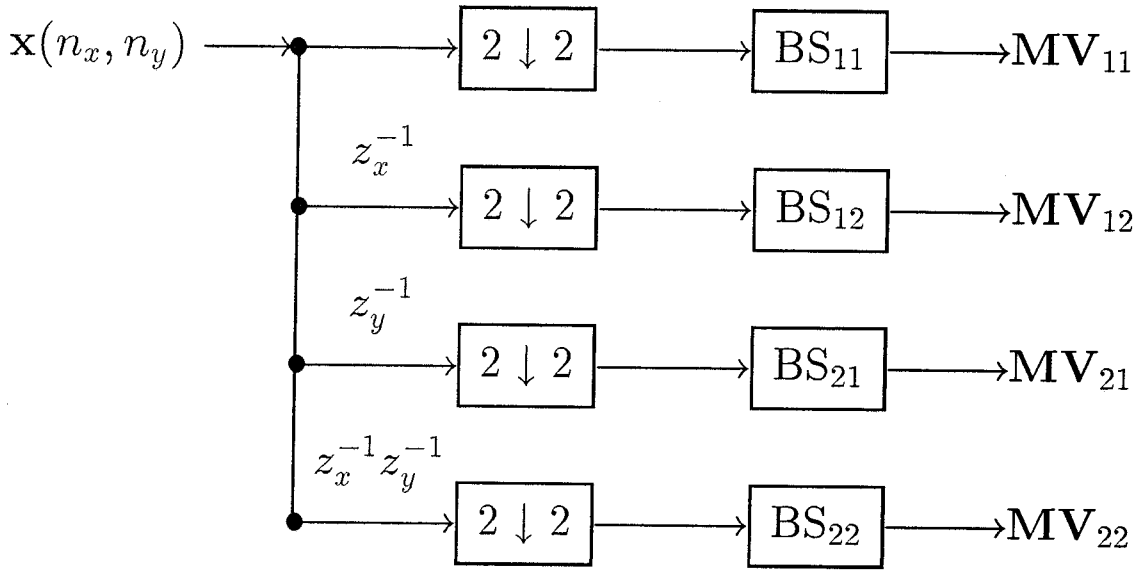


Figure 5.3.1 2-channel subsampled block search.

Example 5.3.1: Figure 5.3.2 shows the comparison of the estimation accuracy of three schemes: FSA, TSS, and 2-channel MBS, using pingpong sequence as the signal source. The PSNR measures the accuracy of estimated image frames compared to the original ones. Two frames of the pingpong sequence are shown in Figure 5.3.3. In FSA and TSS, we set the maximum displacement and block size to be 8 and 8×8, respectively, and 4 and 4×4 in the other one. We can see that the proposed MBS scheme can actually get the best estimation, whereas the complexity is only one-fourth of that of full search. As expected, TSS has the worst performance compared to these two schemes because of the fewer points it searches. To illustrate the benefit provided by motion compensation, we also include the curve corresponding to the case without motion compensation. We just subtract the signals between two frames. It can be seen that one can gain a great deal by applying a motion-compensation scheme.

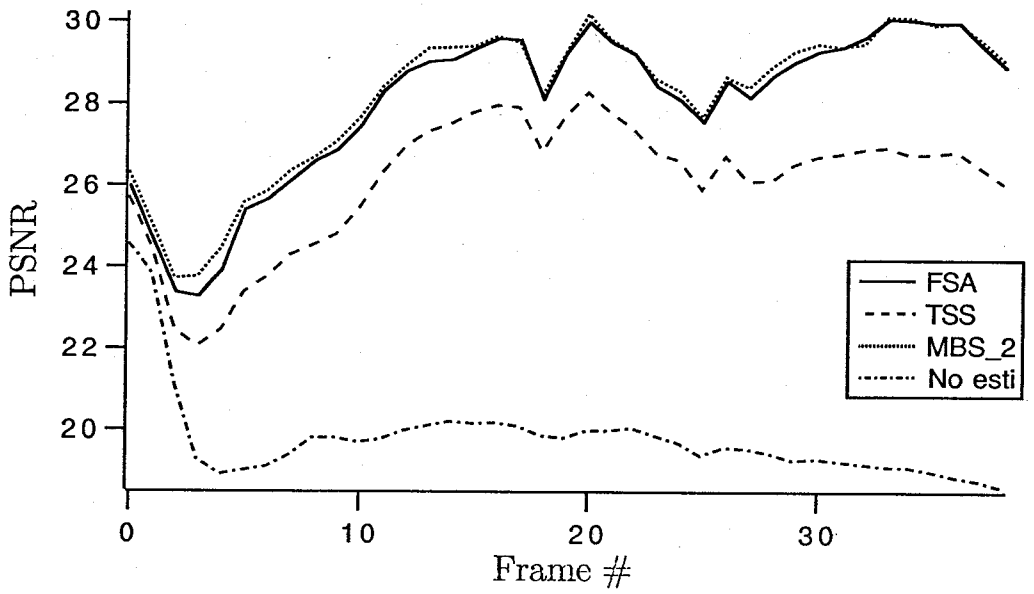


Figure 5.3.2 Comparison of FSA, TSS, and 2-channel MBS.

The above example demonstrates the superiority of this simple MBS algorithm. We need more overhead now, however, to transmit the motion vectors. For the FSA

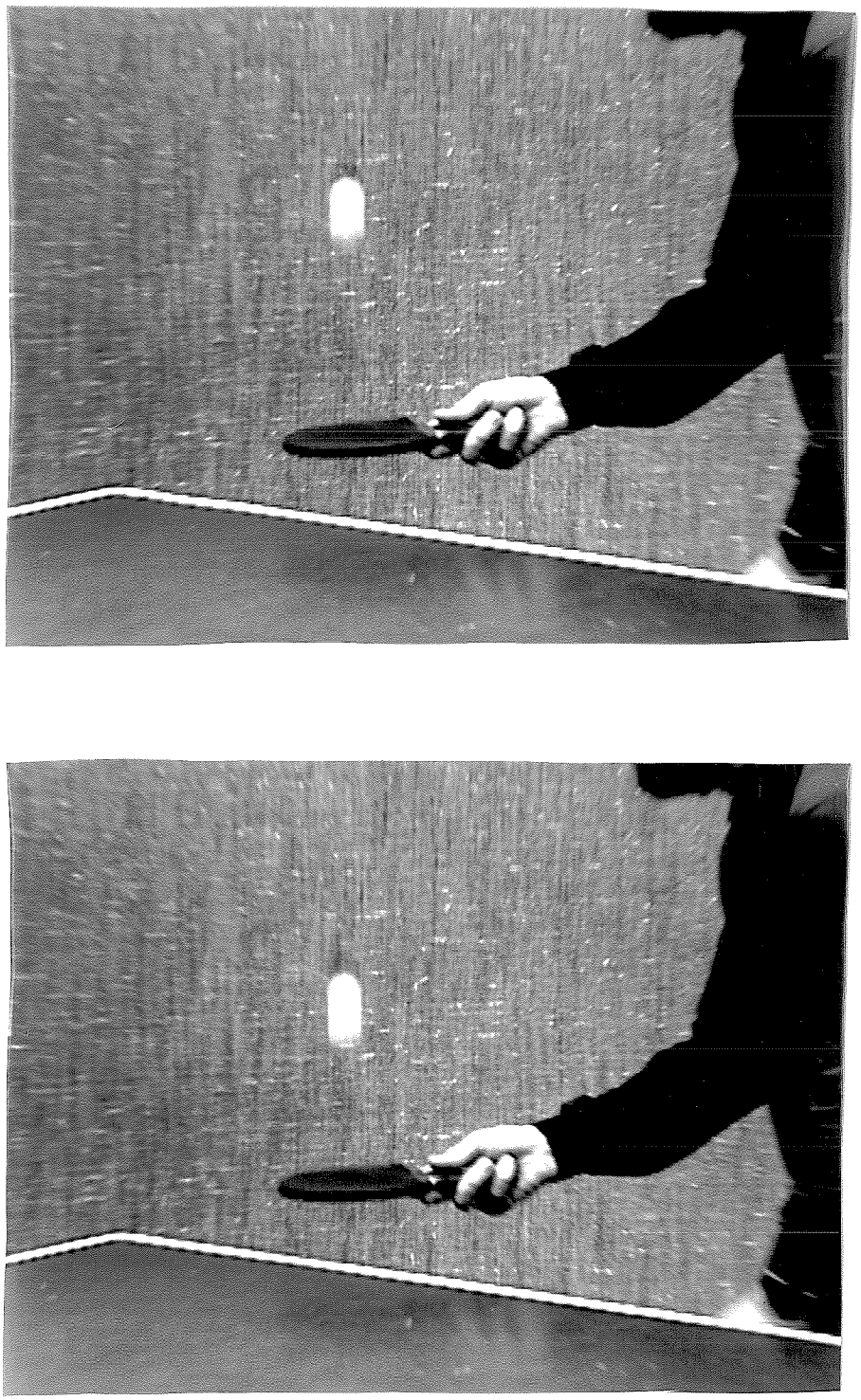


Figure 5.3.3 Two original frames of the pingpong sequence.

and TSS cases, we need the overhead of $\frac{XY}{N^2} \times \lceil \log_2(2d+1)^2 \rceil$ bits totally (in the following, we ignore the constant of $\frac{XY}{N^2}$), whereas in the 2-channel MBS case, we need $4 \times \lceil \log_2(d+1)^2 \rceil$, where $\lceil x \rceil$ is the smallest integer no less than x. We propose the following solutions to this problem:

(a) Entropy code the motion vectors. The redundancy of the motion vectors in the MBS case is expected to be higher than that in the other two cases. This is because the motion information among different bands is highly correlated. Thus, if we entropy code the motion vectors, we should be able to save more bits than in FSA and TSS. For example, for the pingpong sequence, the average redundancy of MV's in FSA, TSS, and 2-channel MBS is 1.2, 1.1, and 2.1 bits, respectively.

(b) Take advantage of the high correlation of the motion vectors among different subimages. Because we subsample the image signals first, then perform block search, there is a very strong correlation of motion vectors at the same block positions in all subimages. To use this correlation, we propose the following variants:

(b.1) At block position (i,j), compute the motion vectors of all subimages at the same positions. Then use the one that results in minimum total distortion as the motion vector for all blocks; i.e.,

$$(m_{l_i}, n_{l_j}), \text{ for some } 0 \le l < 4 \text{ such that}$$

$$\sum_{k=1}^{4} D_{k_{(i,j)}}(m_{l_i}, n_{l_j}) \text{ is minimum.} \tag{5.3.1}$$

Then (m_{l_i}, n_{l_j}) is the motion vector for all of the blocks at position (i,j). Applying this scheme, the overhead is $2 + \lceil \log_2(d+1)^2 \rceil$, which is approximately equal to that of FSA, while the complexity is reduced by an approximate factor of 4. Figure 5.3.4 shows the simulation results of this proposed scheme.

We see that the performance is degraded slightly, but still better than that of TSS.

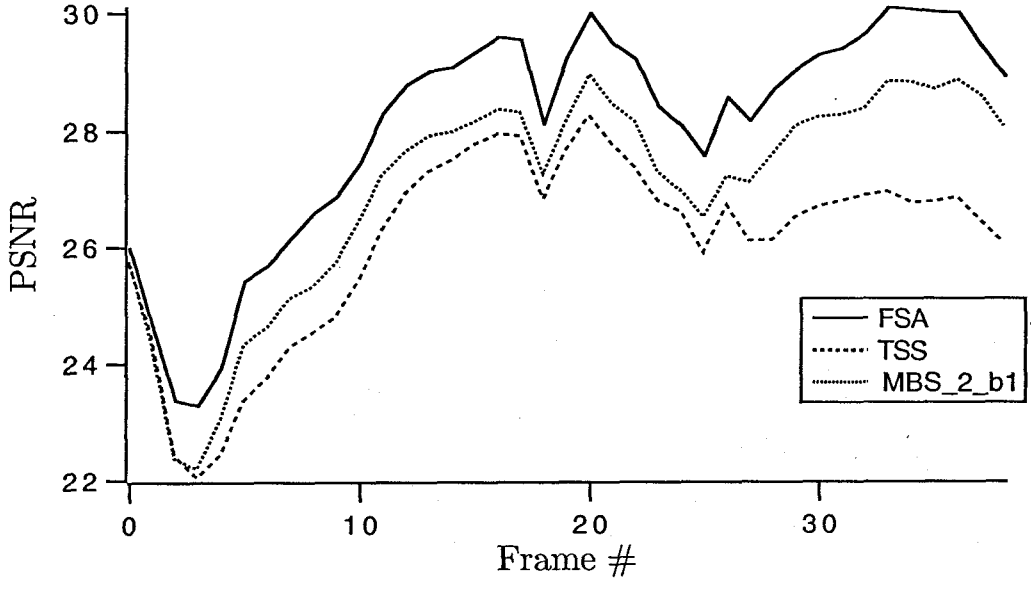


Figure 5.3.4 Comparison of FSA, TSS, and 2-channel MBS (b.1).

(b.2) At block position (i, j), compute the motion vector of one band, e.g., band 0, first, (m_{0_i}, n_{0_j}). For the motion vectors of the other bands at the same position, obtain them from the window $\{(x, y) | i + m_{0_i} - d_1 < x < i + m_{0_i} + d, j + n_{0_j} - d_1 < y < j + n_{0_j} + d_1\}$ in every band, where d_1 is a positive number much less than the maximum displacement d of band 0. In other words, we first use the motion vectors of band 0 to represent approximately those of the other bands. Then we refine these approximated motion vectors in a small window centered at $(i + m_{0_i}, j + n_{0_j})$ for every i, j and every band. The overhead now is $3 \times \lceil \log_2(2d_1 + 1)^2 \rceil + \lceil \log_2(d + 1)^2 \rceil$. Figure 5.3.5 shows the simulation results of this scheme. Here we set $d = 8$ and $d_1 = 1$. Compared to FSA, the computational gain factor of 10.7 is obtained.

(b.3) Use the motion vectors of some subimage to represent approximately those of

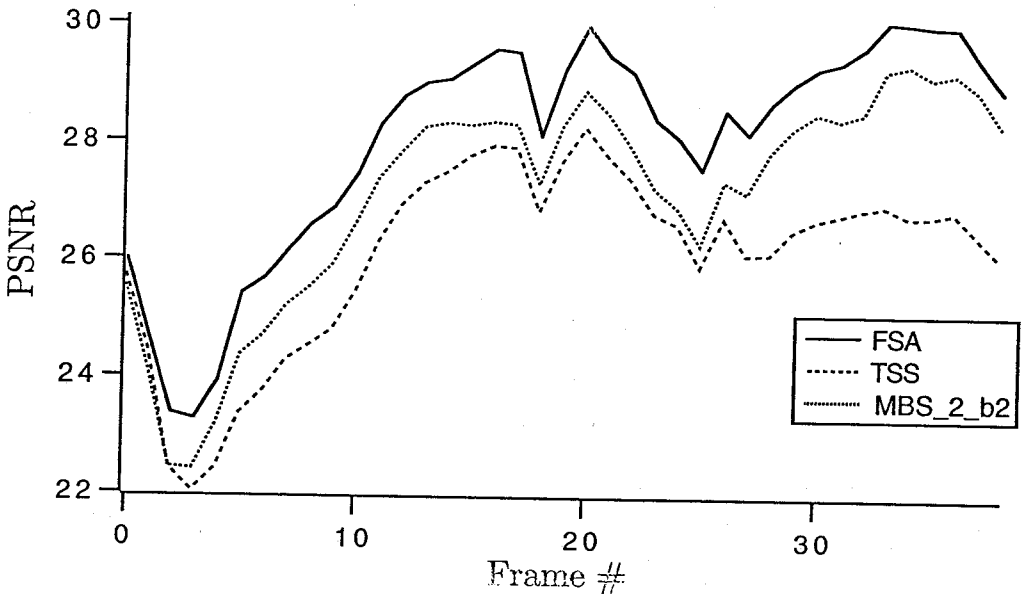


Figure 5.3.5 Comparison of FSA, TSS, and 2-channel MBS (b.2).

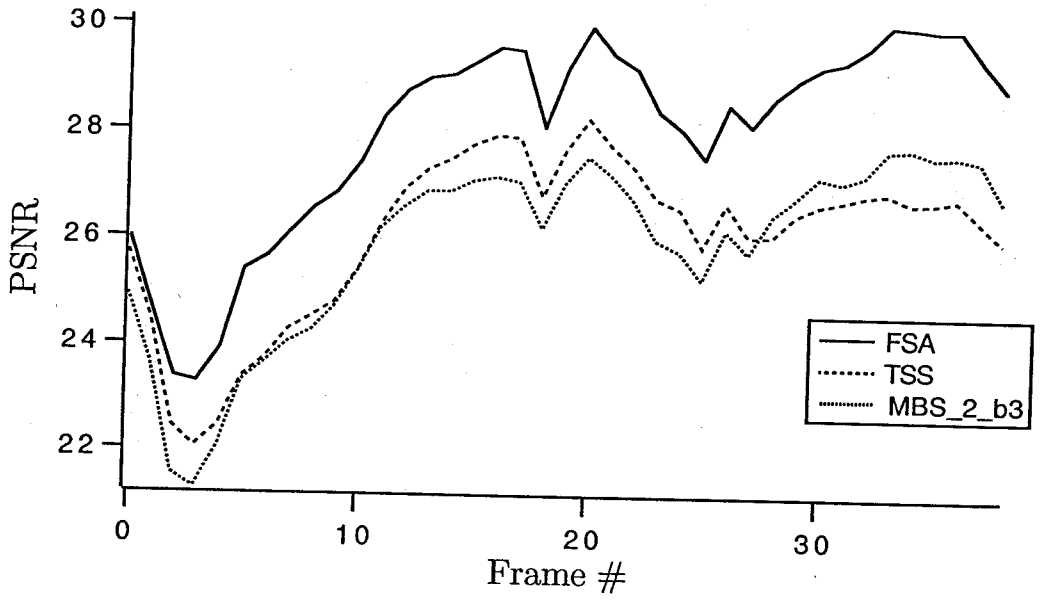


Figure 5.3.6 Comparison of FSA, TSS, and 2-channel MBS (b.3).

the other ones without computing theirs. Now we have the overhead reduced to $\lceil \log_2(d+1)^2 \rceil$ and complexity reduced by a factor of 16. Of course, we will lose some accuracy here. Because of the high correlation of the motion vectors among different subimages, however, we will not sacrifice much precision, and will save a great deal of overhead and computations. Figure 5.3.6 shows the simulation results of this scheme. We can see that the performance of this proposed scheme is close to, or sometimes better than, that of TSS, and the complexity and overhead are greatly reduced.

Figure 5.3.7 shows the comparison of schemes (b.1), (b.2), and (b.3). We can see that the performance of (b.1) and (b.2) is very close. Basically, (b.1) is designed to save the overhead, whereas (b.2) is for reducing the computational complexity (and some overhead). Thus, it is expected that these two schemes have close performance. In scheme (b.3), a great deal of complexity and overhead are reduced. Even though it has the worst estimation accuracy, it is worthwhile to apply this scheme to exploit the temporal redundancy.

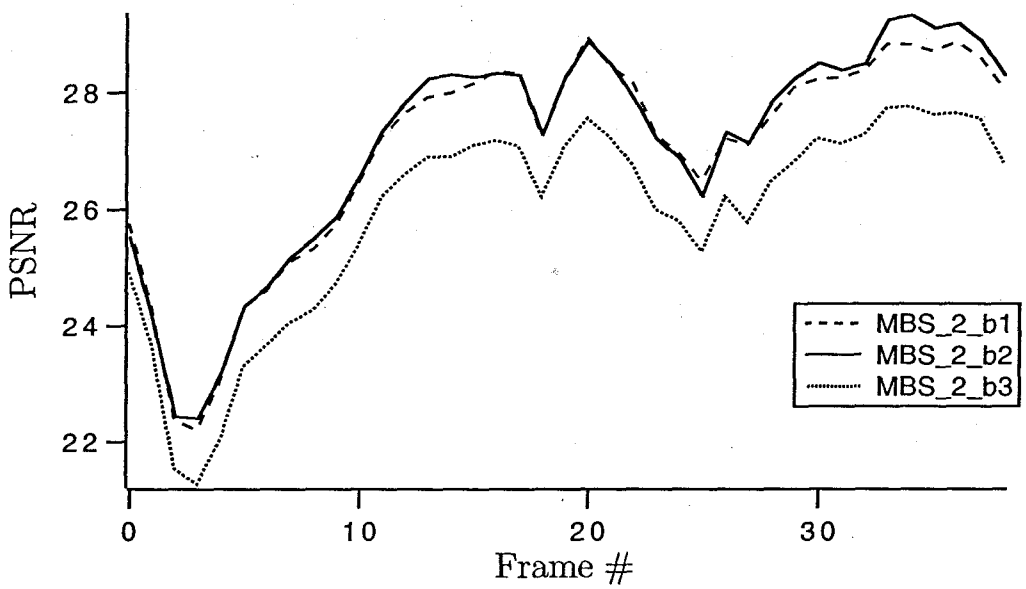


Figure 5.3.7 Comparison of 2-channel MBS (b.1), (b.2), and (b.3).

Note that, although the complexity of schemes (a) and (b.1) is higher than

that of schemes (b.2) and (b.3), they can, however, be implemented by parallel-processing. This means that we can concurrently perform the motion-estimation operations in all subimages, thus greatly reducing the required processing time so that it is the same as in scheme (b.3).

As stated, it is intuitive that the performance of scheme (b.3) is worse than that of (b.1) and (b.2), especially when the number of channels becomes larger. To decrease the degradation so as to save the complexity and overhead, we can pre-process the signals first so that the energy is compacted to one channel [Vaidyanathan, 1993], [Zhang and Zafar, 1992], [Zafar et al., 1993]. Then one estimates the motion vectors of that band to approximately represent those of the other bands. Thus, we can guarantee that the distortion generated in the other bands is much less, as is the total distortion. The generalized multiscale block-search algorithm, which is discussed in the next section, can achieve this goal.

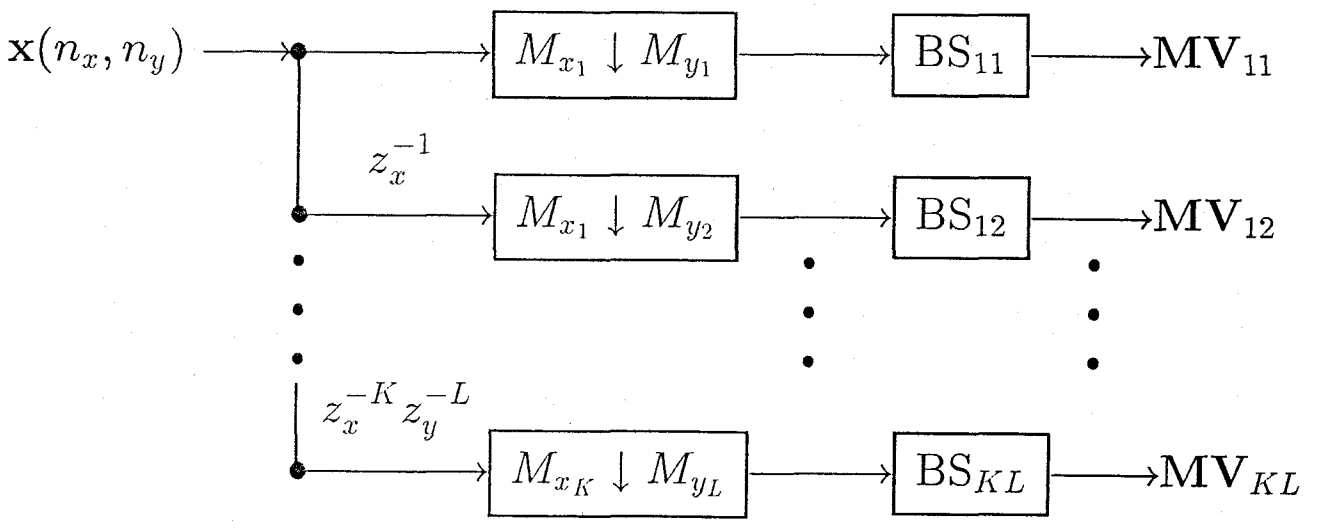


Figure 5.3.8 Block diagram of a KL-channel MBS.

The above shows several approaches of the simplest 2-channel MBS. It is straightforward to extend these schemes to M-channel systems with minor modifications.

Figure 5.3.8 is a block diagram of this generalized case. Now the maximum displacement becomes $\frac{d}{M_{x_k}}$ and $\frac{d}{M_{y_l}}$ in the x and y directions, respectively, and the block size becomes $\frac{N}{M_{x_k}} \times \frac{N}{M_{y_l}}$ in the klth subimage. The computational gain over FSA on the original frames, therefore, is

$$\mathcal{C}_1 = \frac{(2d+1)^2}{\sum_{k=1}^{K}\sum_{l=1}^{L} \frac{1}{M_{x_k} \times M_{y_l}} (2\frac{d}{M_{x_k}} + 1)(2\frac{d}{M_{y_l}} + 1)} \tag{5.3.2}$$

for schemes (a) and (b.1),

$$\mathcal{C}_2 = \frac{(2d+1)^2}{\frac{1}{M_{x_k} \times M_{y_l}} (2\frac{d}{M_{x_k}} + 1)(2\frac{d}{M_{y_l}} + 1) + \frac{(KL)^2 - 1}{(KL)^2}(2d_1 + 1)^2} \tag{5.3.3}$$

for scheme (b.2), and

$$\mathcal{C}_3 = \frac{(2d+1)^2}{\frac{1}{M_{x_k} \times M_{y_l}} (2\frac{d}{M_{x_k}} + 1)(2\frac{d}{M_{y_l}} + 1)} \tag{5.3.4}$$

for scheme (b.3), if we use band k, l as the reference in (b.2) and (b.3), which can be large numbers. For example, for a uniform 4-channel MBS case (resulting in 16 subimages), $\mathcal{C}_1 = 11.56$, $\mathcal{C}_2 = 28.9$, and $\mathcal{C}_3 = 184.96$. The overhead is now

$$\mathcal{H}_1 = \sum_{k=1}^{K}\sum_{l=1}^{L} \left\lceil \log_2 \left(\frac{2d}{M_{x_k}} + 1 \right) \left(\frac{2d}{M_{y_l}} + 1 \right) \right\rceil \tag{5.3.5}$$

for scheme (a),

$$\mathcal{H}_2 = \lceil \log_2(KL) \rceil + \left\lceil \log_2 \left(\frac{2d}{M_{x_k}} + 1 \right) \left(\frac{2d}{M_{y_l}} + 1 \right) \right\rceil \tag{5.3.6}$$

for scheme (b.1),

$$\mathcal{H}_3 = \left\lceil \log_2 \left(\frac{2d}{M_{x_k}} + 1 \right) \left(\frac{2d}{M_{y_l}} + 1 \right) \right\rceil + (KL - 1) \times \lceil \log_2(2d_1 + 1)^2 \rceil \tag{5.3.7}$$

for scheme (b.2), and

$$\mathcal{H}_4 = \left\lceil \log_2 \left(\frac{2d}{M_{x_k}} + 1 \right) \left(\frac{2d}{M_{y_l}} + 1 \right) \right\rceil \tag{5.3.8}$$

for scheme (b.3). Note that these numbers all represent the data amount for the raw information; i.e., before entropy encoding.

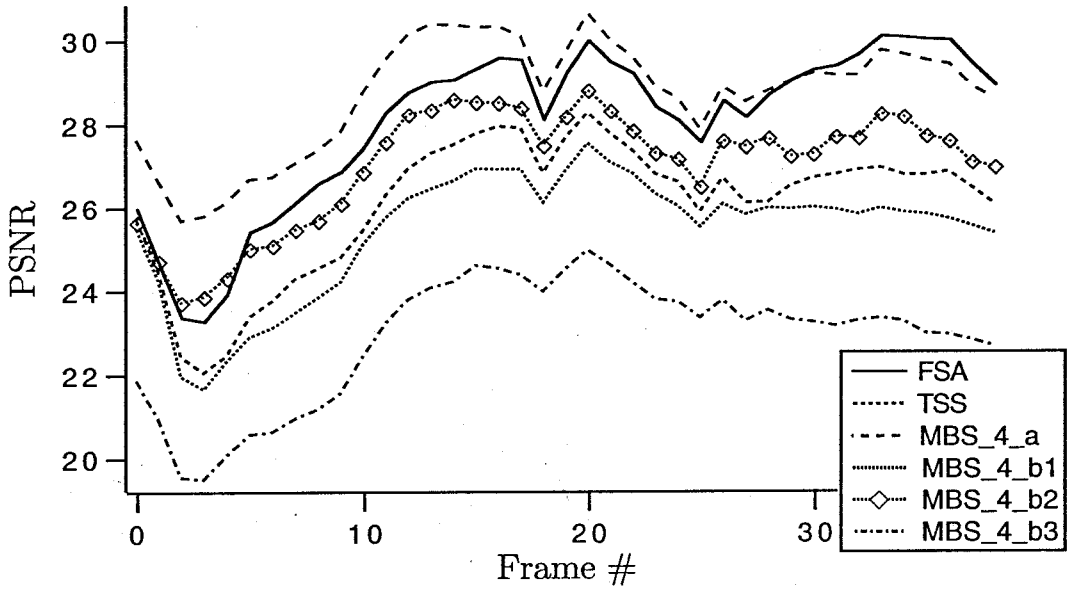


Figure 5.3.9 Comparison of FSA, TSS, and 4-channel MBS.

Example 5.3.2: Figure 5.3.9 shows the comparisons of the following schemes: FSA, TSS, 4-channel MBS of scheme (a), (b.1), (b.2), and (b.3). Note that, in scheme (b.3), because we use the motion vectors of one band to represent those of the other 15 bands, it is expected that it has the worst performance among these schemes. The complexity now is proportional to 1.5625, however, compared to 289 in FSA, which is reduced by a factor of 185, and its performance is much better than that without motion compensation (see Figure 5.3.3). Also the overhead is less than half of that of FSA. This also demonstrates its feasibility.

5.4. Generalized Multiscale Block Search

The schemes in the previous section provide methods of how to subsample the

signals first to reduce the maximum displacement and save computations without sacrificing performance. Besides this, we can also take advantage of the non uniform distribution of the image signals' power spectrum to achieve more savings. As stated, the power spectrum of image signals tends to be non uniform [Woods and O'Neil, 1986], [Woods, 1991]. Generally speaking, lower-frequency parts have higher energy, whereas higher-frequency parts have lower energy. In order to use this property in motion compensation, we propose the scheme of Figure 5.4.1. We first pass an image frame through a filter bank to decompose it into several subimages in different scales with different variance, then apply scheme (a), (b.1), (b.2), and (b.3) of the previous section to perform motion estimation. Of course, all of the arguments of the previous section still hold here. To take advantage of the non uniform property, however, we have the following additional schemes and observations:

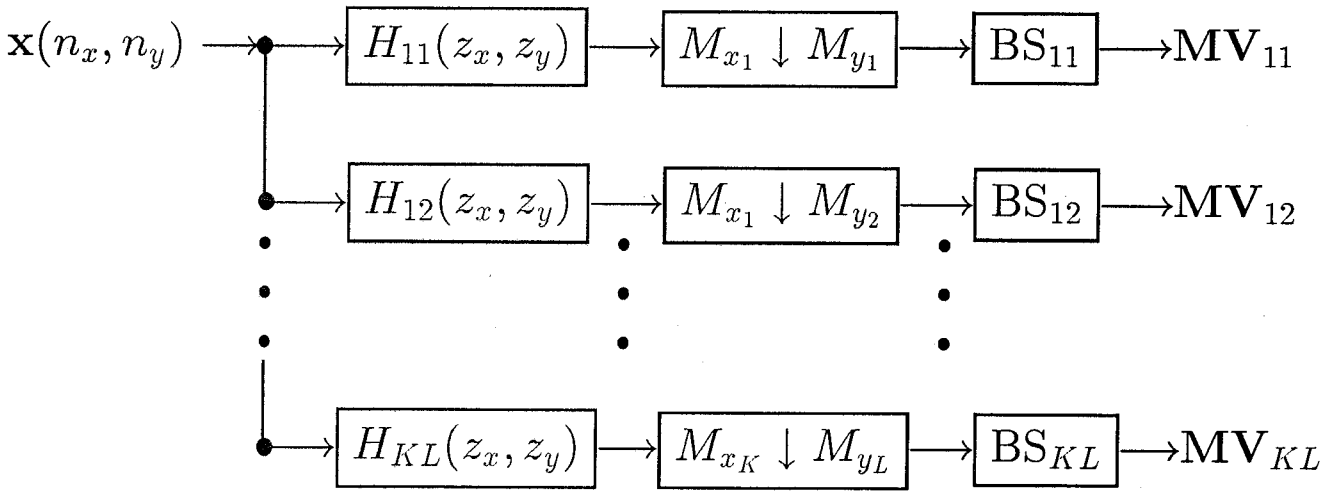


Figure 5.4.1 Block diagram of generalized multiscale block search.

(a.1) Apply different motion-estimation schemes with various accuracy in different subimages, depending on the variance in that band. For example, because of the high variance in low-low band, we apply FSA; in the other bands, we

can apply some fast algorithms, such as TSS, TDL, CDS, and so on. The idea behind this scheme is similar to the optimal bit-allocation strategy [Soman and Vaidyanathan, 1991], [Vaidyanathan, 1993] of intraframe subband coding. Given a certain total complexity, we should "allocate the complexity" to every band so that the total distortion is minimized. It can be shown [Soman and Vaidyanathan, 1991], [Vaidyanathan, 1993], [Jayant and Noll, 1984] that, for a paraunitary (PU) filter bank system, the total distortion is minimized if and only if the distortion in all bands is equalized. For the lower-frequency subimage, then, we should apply a more accurate estimation scheme because of its higher variance. There is no explicit formula, however, to express the input-output relationship of the signal variance for motion compensation because it also strongly depends on how the objects move. An optimal allocation equation cannot be obtained here. Figure 5.4.2 shows a simple example of the proposed scheme. Here we

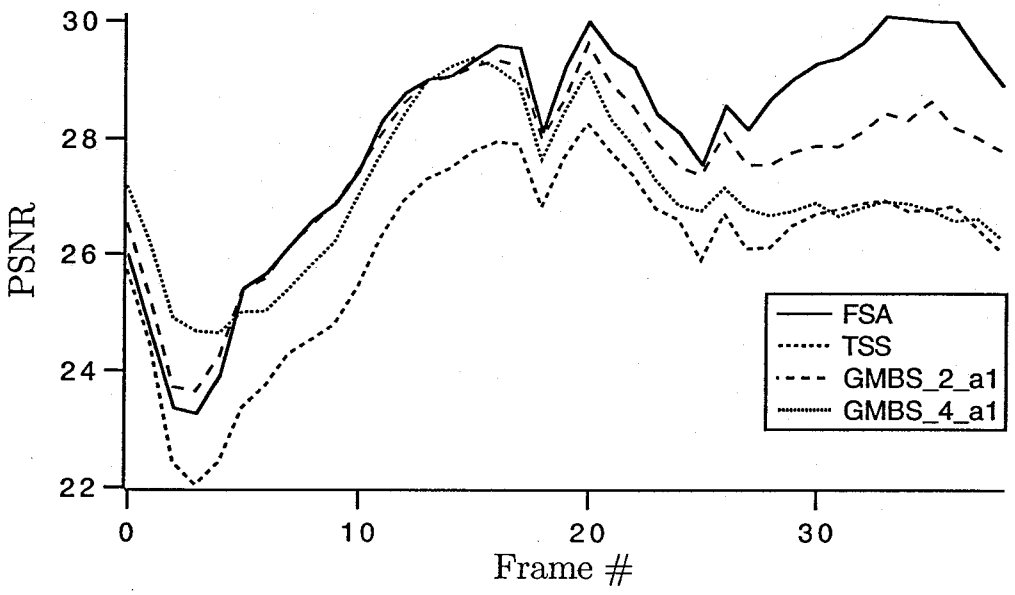


Figure 5.4.2 Comparison of FSA, TSS, 2-channel GMBS (a.1), and 4-channel GMBS (a.1).

apply uniform 4-channel generalized multiscale block search (GMBS). We apply FSA in the low-low band and TSS in the other 15 bands. We can see that the precision is very close to that of FSA, whereas the complexity is greatly reduced (proportional to 33, or a gain of 8.76). This demonstrates the advantage of the proposed scheme.

(b.3) As stated, because most of the energy is compacted to the low-low subimage, we use the motion vectors of this band to represent those of the other bands. Comparing to scheme (b.3) of the previous section, one can obtain a more accurate estimation because the distortion resulting from the other estimated bands is much less. Figure 5.4.3 illustrates this fact. Here we compare the 4-channel case. We can see that the performance of the GMBS (b.3) scheme is indeed better than that of MBS (b.3).

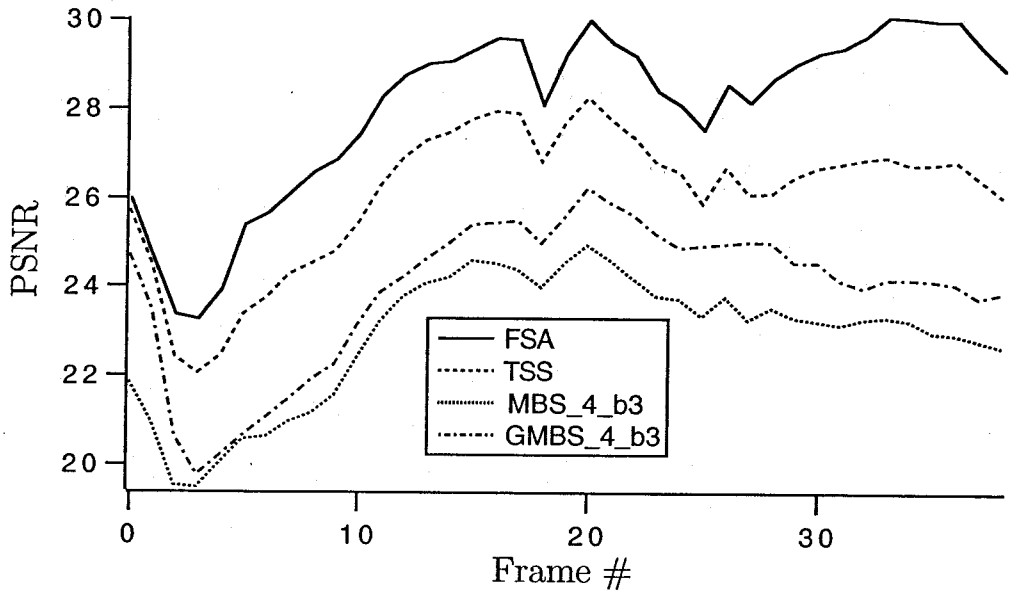


Figure 5.4.3 Comparison of FSA, TSS, and 4-channel MBS (b.3), GMBS (b.3).

(c) In some high-frequency subimages, the signal variance may be very small. For these subimages, we may not need to apply motion compensation because it is not clear whether the signal variance can be reduced after motion compensation.

Even if it can be reduced by a small amount, it is not worthwhile to spend such high complexity for motion estimation to achieve little improvement. Combining this property with scheme (a.1), we have the following approach:

- Compute the signal variance σ_i^2 of every subimage first.
- If $\sigma_i^2 \in R_k$, then apply motion-estimation scheme k in subimage i, where scheme k can be any motion-estimation scheme, or without motion compensation. As mentioned before, typically speaking, if $\sigma_i^2 > \sigma_j^2$, then the accuracy of the scheme used in band i is no less than that in band j.

An additional advantage of this GMBS scheme is that it can greatly smooth out the blocking effects generated by the block search. As stated in Section 5.2., BSA tends to generate very strong blocking effects in the estimated images, which may greatly degrade the image quality. Applying GMBS, we have a set of synthesis filters in the receiving end to smooth out the annoying effects, thus improving the picture's visual quality–a great advantage.

Note that, although we have proposed methods of constructing the block size and maximum displacement, we can still change these numbers in different subimages. The motivation for setting these parameters in the proposed schemes is from the viewpoint of subsampling and trying to compare the accuracy with FSA. We can, of course, have different parameters in different subimages, thus providing more flexibility. For example, in schemes (a) and (a.1), we can set the block size to be smaller and maximum displacement to be larger in the lower-frequency band.

Fine points about the filter banks

Next we mention some points about the construction of the filter banks. For simplicity, here we apply separable filters; i.e.,

$$H_{kl}(z_x, z_y) = H_k(z_x)H_l(z_y), \qquad 1 \le k \le K, 1 \le l \le L. \tag{5.4.1}$$

Herein we propose to use *linear-phase* filter banks [Nguyen and Vaidyanathan, 1989], [Soman et al., 1993], [Vaidyanathan, 1993] to avoid phase distortion in subimages. This means that motion information can still be retained in subbands. We can still preserve the strong correlation of motion vectors among different subimages, therefore, which will greatly benefit schemes (b.1), (b.2), and (b.3). We apply the algorithms developed in [Nguyen and Vaidyanathan, 1989], [Soman et al., 1993] to obtain the filter coefficients for the 2- and 4-channel filter banks, respectively. Tables 5.4.1 and 5.4.2 show these coefficients for the 2- and 4-channel cases, respectively.

n	$h_0(n)$	$h_1(n)$	$f_0(n)$	$f_1(n)$
0	0.0220851391	0.0008264153	−0.0008264153	0.0220851391
1	0.0240330876	0.0008993066	0.0008993066	−0.0240330876
2	−0.1040810278	−0.1037503319	0.1037503319	−0.1040810278
3	0.1028960652	−0.1048127948	−0.1048127948	−0.1028960652
4	0.4766289373	0.4687792213	−0.4687792213	0.4766289373
5	0.4766289373	−0.4687792213	−0.4687792213	−0.4766289373
6	0.1028960652	0.1048127948	−0.1048127948	0.1028960652
7	−0.1040810278	0.1037503319	0.1037503319	0.1040810278
8	0.0240330876	−0.0008993066	0.0008993066	0.0240330876
9	0.0220851391	−0.0008264153	−0.0008264153	−0.0220851391

Table 5.4.1 Coefficients of the two-channel, linear-phase filter banks.

n	$h_0(n)$	$h_1(n)$	$h_2(n)$	$h_3(n)$
0	−0.0915848091	−0.1335739046	−0.1335739046	−0.0915848091
1	0.1335739046	0.0915848091	−0.0915848091	−0.1335739046
2	0.3892334104	0.5676861405	0.5676861405	0.3892334104
3	0.5676861405	0.3892334104	−0.3892334104	−0.5676861405
4	0.5676861405	−0.3892334104	−0.3892334104	0.5676861405
5	0.3892334104	−0.5676861405	0.5676861405	−0.3892334104
6	0.1335739046	−0.0915848091	−0.0915848091	0.1335739046
7	−0.0915848091	0.1335739046	−0.1335739046	0.0915848091

Table 5.4.2 Coefficients of the four-channel, linear-phase PU analysis filter bank; the coefficients of synthesis filter bank are $f_k(n) = h_k(N - n)$.

5.5. Discussions and Summary

We have presented several interframe compression schemes, using generalized multirate techniques. The motivations, principles, and simulation results have been explained to support the superiority. These schemes not only reduce the computational complexity, but also improve the resulting picture quality. Also, these schemes provide high flexibility in processing data. One can adaptively apply different strategies to different subimages, depending on the signal characteristics and applications. Furthermore, we can achieve pure progressive transmission by applying this scheme, as subband coding itself is intrinsically a progressive-type transmission method. Comparing to the traditional approaches that apply motion compensation first to the original frames, then apply subband coding to the compensated signals, here we can independently process the data in every subband. Upon completing processing of the signals in the low-resolution subimage, one can expand it and start to browse it. When more subimages are received, more details can be obtained. These schemes, therefore, are also suitable for *packet video* [IEEE JSAC, 1989], [IEEE CSVT, 1993].

As mentioned, in order to deal with the uncovered regions, we can incorporate interpolative prediction. This means that we can also apply these schemes to MPEG systems; i.e., after multiscale processing, we can have I, B, and P frames in subimages periodically, then apply DCT or scalar quantization to all of the compensated subimages separately. Herein I, B, and P frames refer to Intrafield, Bidirectionally predicted, and Predicted frames. This can improve the flexibility and performance of MPEG to make it more feasible. Because applying interpolative prediction (B frames) costs more computations, overhead, and buffer for storing more reference frames (and thus results in higher frame delay), however, it is still controversial whether to apply this technique.

The video signals we used in simulations were with high movement. If we use a video-conferencing-type signal source, the gain of applying motion compensation will be much higher. We only need to spend very few bits in intraframe compression.

In this case, we apply only the principles of block search to every band. We can, of course, apply a more complex motion-estimation scheme to subbands because these subimages still represent meaningful images, and the motion information is still strongly related among subbands. We even speculate that the correlation of the motion vectors among different bands obtained by these better estimation schemes may be higher, as the motion information in every band is more accurately detected. If there is no phase distortion in subbands, the motion information among different subbands is highly correlated. We can, therefore, gain even more in schemes (b.1), (b.2), and (b.3).

Motion compensation is still an open field, and deserves much more research. It is believed that it is the motion-compensation technique that has the highest potential further to greatly compress the video signals.

Chapter 6

Linear Multiscale Techniques for Video Compression

6.1. Motivation and Block Diagram

In this chapter, we develop a video-compression scheme by combining the intra- and interframe algorithms discussed in the previous chapters. As stated, a video-compression system is basically composed of three parts: an interframe compressor, intraframe compressor, and lossless data compactor. We first apply motion compensation to the video signals to take advantage of the temporal redundancy to decrease the signal variance. We then apply intraframe compression to the estimated error signals by using the spatial redundancy and properties of HVS to further reduce the number of bits. Finally, we apply an entropy encoder to losslessly decrease the bit amount.

It has been demonstrated in the previous two chapters that many computations can be saved by applying VQ and motion compensation to the subbands. Based on this, we propose the scheme of *generalized multiscale motion compensation with VQ*. The block diagram is shown in Figure 6.1.1. We first pass the one frame of signals through an analysis filter bank, then apply motion compensation to every subimage. Finally, we apply PCVQ to the compensated signals in every subband. Note that, although we perform motion compensation after filtering, there is almost no interblock correlation among the blocks in different subbands. We can, then, still take advantage of this and apply VQ to the blocks in each frequency band.

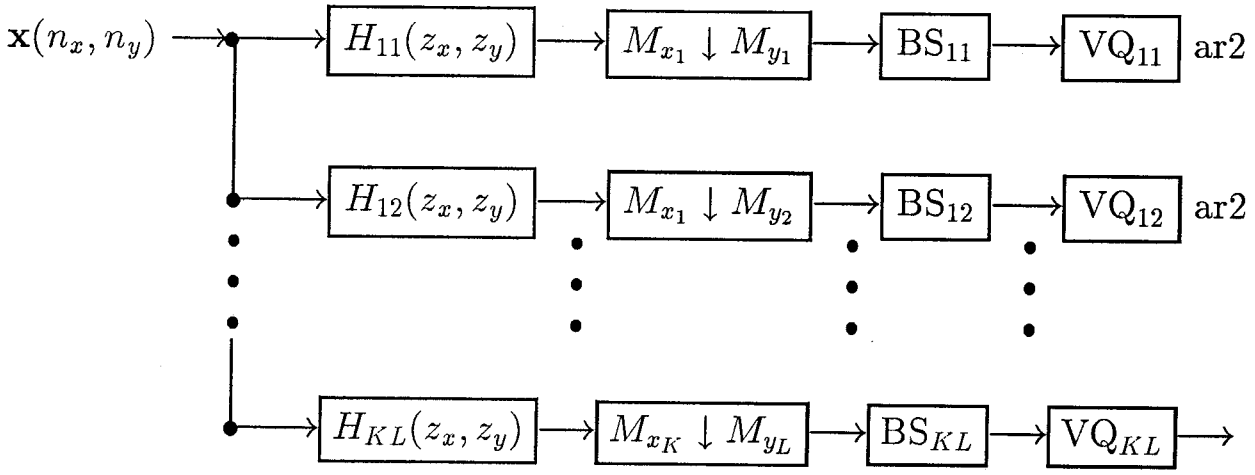


Figure 6.1.1 Block diagram of generalized multiscale motion compensation with VQ.

6.2. Simulation Results

We now demonstrate and investigate the quality of the compressed pictures after applying the whole system — subbanding, motion compensation, VQ, and their reverse-decoding processes. Herein we have assumed that the transmission of motion vectors and the indices of codevectors are free from error (lossless channel).

As mentioned in Section 4.4, deciding the number of bits (codebook size) allocated to one frequency band is based on the signal statistics in that band. Before applying PCVQ, therefore, we need to investigate the signal characteristics after the subbanding and motion-compensation operations. Herein we consider only the four-channel case. The coefficients of the filter bank are shown in Table 5.4.2. Table 6.2.1 shows the variance of the signals in every band before motion compensation, after FSA in every band, and after TSS in every band, respectively, and the codebook size allocated to each band in the FSA case. (In simulations, we will show only results of this case.) Figure 6.2.1 shows the histograms of signals after applying FSA, where the number n in the first column represents the value of the signals in the range of $(n, n+1]$ ($a \in (n, n+1]$ means $n < a \leq n+1$), whereas the numbers

in the other columns stand for the amount of signals in the corresponding range. From Table 6.2.1 and Figure 6.2.1, we can see that, after motion compensation, the signal variances in all bands are decreased by a large amount. Almost all of the signals in all bands (except for band 0) have values in the range of $(-1, 1)$. This demonstrates the fact that we can save a large number of bits by applying motion compensation first.

	0	1	2	3	4	5
Before MC	994.176	20.533	17.265	5.656	67.409	7.222
After FSA	29.255	4.787	4.605	1.486	11.026	2.235
After TSS	57.702	7.738	7.124	2.364	23.652	3.436
Codebook size	128	4	1	1	16	1

	6	7	8	9	10	11
Before MC	6.974	2.706	24.301	5.283	12.308	3.494
After FSA	2.256	0.831	4.932	1.767	3.020	0.936
After TSS	3.417	1.321	9.165	2.742	5.015	1.548
Codebook size	1	1	1	1	1	1

	12	13	14	15
Before MC	40.420	10.579	18.880	6.092
After FSA	11.823	3.728	4.520	1.424
After TSS	22.244	6.125	7.808	2.438
Codebook size	32	1	1	1

Table 6.1.1 Variances before and after motion compensation (MC) and codebook size in subbands.

From Table 6.1.1, we can also see that the variance of the signals in the low-low band remains the highest after motion compensation; therefore, we still allocate most bits to this band. Remember that the optimal bit-allocation strategy is to make the quantization error in every band equal (actually, approximately equal

Value	BD0	BD1	BD2	BD3	BD4	BD5	BD6	BD7	BD8	BD9	BD10	BD11	BD12	BD13	BD14	BD15
-81	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-80	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-79	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-77	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-76	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-75	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-74	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-73	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-72	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-71	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-69	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-68	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-67	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-66	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-65	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-64	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-63	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-62	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-61	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-60	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
-59	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-58	2	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
-57	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-55	1	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0
-54	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-53	3	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
-52	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-51	5	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0
-50	1	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0
-49	3	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0
-48	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-47	1	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0
-46	2	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
-45	3	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0
-44	5	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
-43	4	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0
-42	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-41	5	0	0	0	1	0	0	0	0	0	0	0	3	0	0	0
-40	1	0	0	0	0	0	0	0	1	0	0	0	2	0	0	0
-39	3	0	0	0	0	0	0	0	1	0	0	0	2	0	0	0
-38	7	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0
-37	6	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
-36	8	0	0	0	1	0	0	0	0	0	0	0	5	0	0	0
-35	12	0	0	0	0	0	0	0	1	0	0	0	3	0	0	0
-34	13	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0
-33	8	0	0	0	0	0	0	0	1	0	0	0	7	0	0	0
-32	15	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
-31	15	0	0	0	1	0	0	0	1	0	0	0	3	0	0	0
-30	12	0	0	0	0	0	0	0	1	0	0	0	5	0	0	0
-29	13	0	0	0	1	0	0	0	1	0	0	0	2	0	0	0
-28	22	0	0	0	3	0	0	0	1	0	0	0	8	0	0	0
-27	10	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0
-26	20	0	0	0	1	0	0	0	0	0	0	0	4	0	0	0
-25	20	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0
-24	16	0	0	0	2	0	0	0	0	0	0	0	6	0	0	0
-23	24	0	0	0	2	0	0	0	2	0	0	0	10	1	0	0
-22	17	1	0	0	2	1	0	0	1	0	0	0	11	1	0	0
-21	20	0	1	0	3	0	0	0	0	0	0	0	7	0	0	0
-20	25	2	0	0	3	0	0	0	2	0	0	0	12	2	1	0
-19	27	1	0	0	5	0	0	0	1	0	0	0	13	2	1	0
-18	30	0	1	0	4	0	0	0	0	0	0	0	19	3	0	0
-17	23	2	0	0	7	0	0	0	3	0	0	0	16	4	1	0

Figure 6.2.1 Histograms of the compensated signals in the subbands.

Value	BD0	BD1	BD2	BD3	BD4	BD5	BD6	BD7	BD8	BD9	BD10	BD11	BD12	BD13	BD14	BD15
-16	30	3	0	0	5	2	0	0	4	0	0	0	22	4	1	0
-15	28	10	1	0	5	1	0	0	6	0	0	0	19	6	1	0
-14	32	11	1	0	7	1	0	0	9	1	0	0	18	5	0	0
-13	47	8	1	0	19	3	0	0	5	3	0	0	25	4	0	0
-12	44	16	3	0	32	3	0	0	11	2	0	0	31	7	0	0
-11	70	29	5	0	26	0	0	0	11	1	0	0	38	13	4	0
-10	89	32	13	1	41	4	0	0	23	6	1	0	36	9	5	0
-9	104	68	21	1	50	17	1	1	43	6	1	0	44	17	10	0
-8	180	74	47	7	102	17	2	1	62	9	4	1	66	36	21	3
-7	230	151	66	9	135	36	3	1	125	21	3	0	94	70	38	6
-6	356	270	129	22	211	73	22	5	163	55	17	0	146	130	106	22
-5	481	351	251	90	350	157	72	17	293	110	71	8	248	248	210	59
-4	716	546	420	219	624	387	235	59	581	347	227	66	488	474	455	252
-3	1184	981	940	735	1047	891	825	401	999	835	832	358	996	1037	971	670
-2	2036	1910	2466	2545	2090	2075	2642	2230	2048	2078	2513	2291	2230	2047	2460	2439
-1	3389	4977	5288	5973	4872	5956	5869	6880	5320	5943	5899	6849	4901	5437	5394	6140
0	3520	5201	5246	6014	4768	5937	5858	6899	5106	6064	5843	6958	5007	5336	5165	6232
1	2219	1989	2487	2448	2014	2078	2553	2182	1985	2199	2658	2172	2216	2132	2436	2341
2	1250	970	975	769	1079	877	766	413	1062	917	799	415	1012	1061	1003	711
3	778	583	383	215	589	375	241	92	595	364	221	70	485	532	467	209
4	519	376	204	92	369	149	75	13	311	154	76	7	238	283	240	82
5	337	221	118	38	227	67	19	6	164	53	24	4	130	123	114	21
6	227	148	62	16	150	43	9	0	100	20	7	0	96	72	45	9
7	184	93	33	5	90	15	5	0	54	7	2	1	67	43	26	3
8	132	54	12	0	59	18	1	0	41	2	1	0	55	27	12	1
9	71	43	11	0	33	6	0	0	18	2	0	0	36	9	6	0
10	61	23	5	1	27	4	1	0	12	0	0	0	22	3	2	0
11	59	16	4	0	20	3	0	0	8	0	0	0	21	4	4	0
12	47	18	4	0	13	0	0	0	2	1	0	0	16	2	1	0
13	30	6	0	0	15	3	0	0	4	0	1	0	27	1	0	0
14	29	5	1	0	11	1	0	0	6	0	0	0	17	1	0	0
15	26	3	1	0	8	0	1	0	2	0	0	0	17	1	0	0
16	19	1	0	0	5	0	0	0	1	0	0	0	11	2	0	0
17	21	2	0	0	9	0	0	0	2	0	0	0	12	0	0	0
18	13	1	0	0	5	0	0	0	1	0	0	0	7	2	0	0
19	20	0	0	0	6	0	0	0	0	0	0	0	17	0	0	0
20	15	2	0	0	7	0	0	0	1	0	0	0	7	2	0	0
21	7	0	0	0	4	0	0	0	2	0	0	0	14	0	0	0
22	10	1	0	0	2	0	0	0	0	0	0	0	7	0	0	0
23	10	0	0	0	4	0	0	0	1	0	0	0	8	1	0	0
24	6	1	0	0	4	0	0	0	0	0	0	0	3	1	0	0
25	9	0	0	0	1	0	0	0	0	0	0	0	4	2	0	0
26	8	0	0	0	3	0	0	0	0	0	0	0	9	0	0	0
27	2	0	0	0	2	0	0	0	0	0	0	0	4	0	0	0
28	3	0	0	0	3	0	0	0	0	0	0	0	3	1	0	0
29	5	0	0	0	4	0	0	0	0	0	0	0	6	0	0	0
30	8	0	0	0	3	0	0	0	0	0	0	0	1	0	0	0
31	9	0	0	0	1	0	0	0	0	0	0	0	2	1	0	0
32	3	0	0	0	2	0	0	0	0	0	0	0	4	0	0	0
33	2	0	0	0	1	0	0	0	1	0	0	0	5	0	0	0
34	5	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0
35	6	0	0	0	2	0	0	0	0	0	0	0	6	0	0	0
36	6	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0
37	7	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0
38	10	0	0	0	0	0	0	0	0	0	0	0	3	1	0	0
39	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
40	4	0	0	0	0	0	0	0	1	0	0	0	2	0	0	0
41	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
42	4	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0
43	4	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0
44	4	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0
45	1	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0
46	6	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0
47	5	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Figure 6.2.1 (Continued)

Value	BD0	BD1	BD2	BD3	BD4	BD5	BD6	BD7	BD8	BD9	BD10	BD11	BD12	BD13	BD14	BD15
48	3	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
49	2	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
50	2	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
51	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
52	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0
53	1	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0
54	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
56	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
57	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
58	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
59	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
60	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
61	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
62	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
63	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
64	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
65	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
66	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
67	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
68	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
69	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
71	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
72	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
73	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
74	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
75	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
76	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
77	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
79	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
80	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
81	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 6.2.1 (Continued)

because the codebook size must be an integer, which makes the quantization errors in different bands not exactly equal). Figure 6.2.2 shows two quantized frames (frames 18 and 19) after applying 4×4 PCVQ to the compensated error signals and then decoding. The corresponding original frames are shown in Figure 5.3.4. Herein we consider only the case of FSA in all subbands. The codebook sizes allocated to the subbands are shown in the fourth row of Table 6.1.1. The values of the PSNR of these two frames are 32.523 and 32.539dB, respectively, and the bits per pixel (bpp) is only 0.07. This means that we can achieve a remarkably high performance (high PSNR as well as good perceptual quality without visible blocking effects compared to those obtained by MPEG and H.261) by using such a low bit rate (compression ratio is over 100:1) and by applying this subband motion

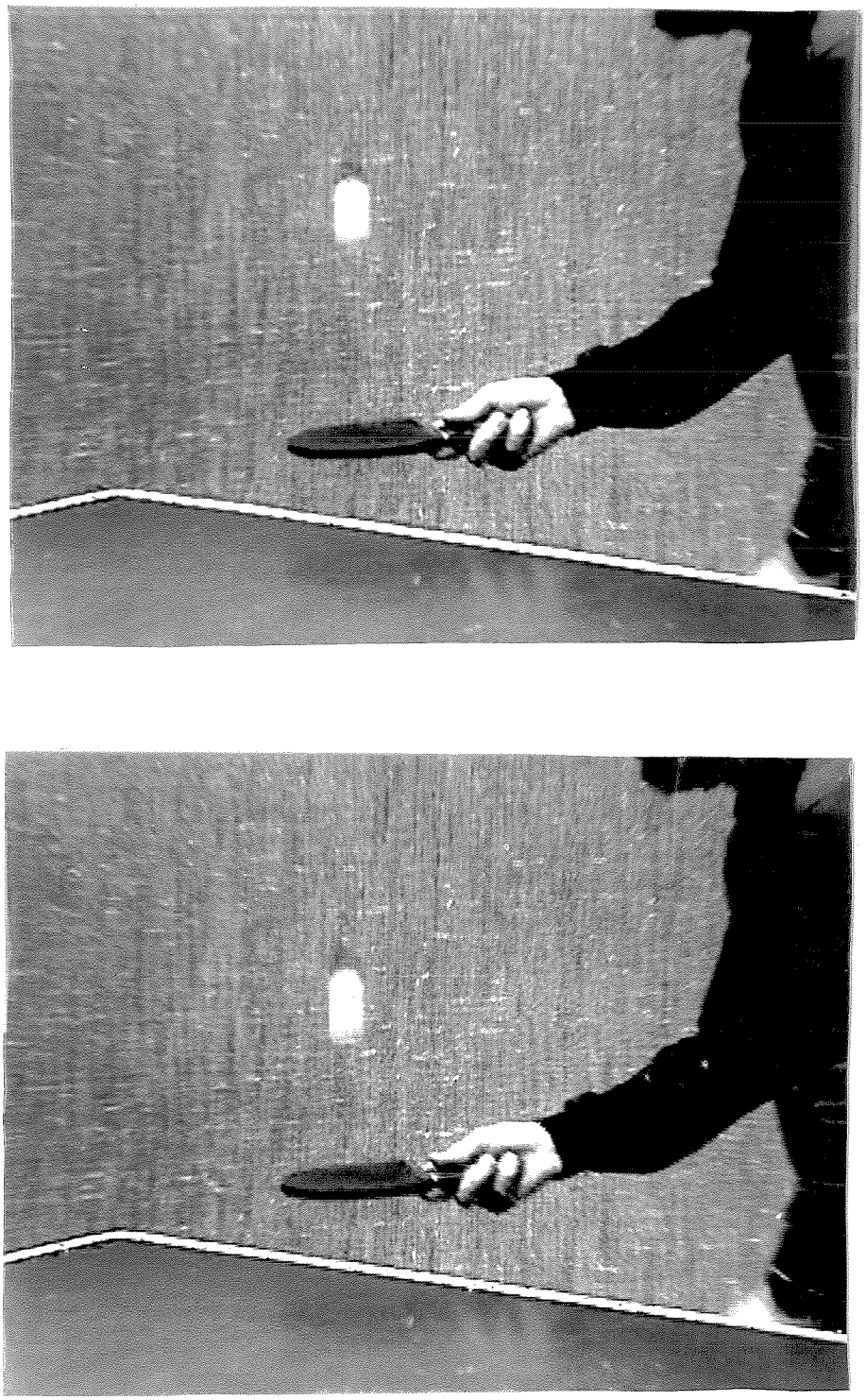


Figure 6.2.2 Two quantized frames of Figure 5.3.3 with 0.1bpp and PSNR of 32.5dB.

compensation with VQ scheme. When we properly include the side information needed for transmitting the motion vectors, the bit per pixel is still low (approximately 0.1bpp, or 80:1 compression).

There are several points we need to mention about the implementation in the simulations:

(1) We have only applied pure forward prediction in motion estimation. B frames are not used. As stated in Chapter 5, if we were to include the B frames, we could expect to obtain an even higher compression ratio by reducing the variance of the compensated signals, especially because, in these 40 frames of pingpong sequence, there are many uncovered regions between frames. This will be at the expense of higher complexity, overhead, frame delay, and buffer size.

(2) Because in these 40 frames, the scene-change did not occur, we have designed a universal codebook for one subimage for all frames (totally, 16 codebooks), except for the first frame. To include the I frames in order to handle the scene-change problem, we should design separate codebooks for the I and P frames (and B frames if necessary).

(3) In our simulations, we have worked on monochrome pictures. If we were to apply the same scheme and algorithms to the color video signals, we would expect to be able to achieve compression ratios as high as 150:1 or more with high picture quality (high PSNR as well as perceptual effect) and reasonable complexity. This is because as mentioned in Chapter 2, Because of the lower sensitivity of HVS to U and V (or I and Q) components, we can subsample these two components by $\boxed{2 \downarrow 2}$ without sacrificing visible distortion. Even the monochrome results demonstrate the feasibility of the proposed scheme in real-time video transmission systems, such as videoconferencing, videophone, and HDTV.

6.3. Concluding Remarks

Conclusively, the principles of the proposed scheme are as follows:

1. We apply VQ to intraframe compression to get a high compression ratio. Because of the characteristics of the compensated signals and the subbanding operations before VQ, the complexity of this part is very low.

2. We apply motion compensation to interframe compression to exploit the temporal redundancy. Because of the fast algorithms as well as the subbanding, the complexity of this part is also low.

3. We use the properties of subbanding in both intra- and interframe compression. In the intraframe case, we use it to remove the interblock correlations of large blocks so that we can apply VQ to smaller blocks and retain the low-bit-rate advantage of a larger block, thus greatly reducing the complexity. In the interframe case, we use the decimation property of subbanding to decrease the maximum displacement and block size, thus greatly decreasing the computations and achieving more accurate estimates. Because we apply analysis filter bank before decimation, the subimages will not suffer from aliasing. Hence, the motion estimation on the subbanded images is still accurate.

We have demonstrated only one implementation and illustrated the simulation results. There are certainly many variants of this proposed scheme. For example, we can apply any interframe compression techniques (with or without subbanding operation) proposed in Chapter 5, followed by VQ, DCT, or any other intraframe technique. The design certainly depends on the desired applications.

Bibliography

This bibliography contains two parts. The first part is an alphabetical list by the authors' names of all of the references cited in the thesis. For specific topics, see the second part of *Selected References by Topic*, which follows the listing of the first part.

Alphabetical Listing by Authors

Ahmad, M. O., and Sundararajan, D., "A Fast Algorithm for Two-Dimensional Median Filtering," *IEEE Trans. Circuits and Systems*, vol. CAS-34, no. 11, pp. 1364–1374, 1987.

Antonioni, M., Barlaud, M., Mathieu, P., and Daubechies, I., "Image Coding Using Wavelet Transform," *IEEE Trans. Image Processing*, vol. IP-1, pp. 205–220, 1992.

Arce, G. R., and McLoughlin, M. P., "Theoretical Analysis of Max/Median Filters," *IEEE Tran. Acoustics, Speech, and Signal Processing*, vol. ASSP-25, no. 1, pp. 60–69, 1987.

Arce, G. R., and Stevenson, R. L., "On the Synthesis of Median Filter Systems," *IEEE Trans. Circuits and Systems*, vol. CAS-34, no. 4, pp. 420–429, 1987.

Argenti, F., Benelli, G., and Mecocci, A., "Source Coding and Transmission of HDTV Images Compressed with the Wavelet Transform," *IEEE J. Selected Areas in Communications*, vol. JSAC-11, no. 1, pp. 46–58, 1993.

Baker, R. L., and Gray, R. M., "Differential Vector Quantization of Achromatic Imagery," in *Proc. International Picture Coding Symposium*, March 1993.

Baseri, R., and Mathews, V. J., "Vector Quantization of Images Using Visual Masking Functions," *Proc. Int. Conf. on ASSP*, vol. 3, pp. 365–368, 1992.

Biemond, J., Looijenga, L., and Boekee, D. E., "A Pel-Recursive Wiener-Based Displacement Estimation Algorithm," *Signal Processing*, vol. 13, pp. 399–412, 1987.

Bradley, J. N., Stockham, T. G., and Mathews, V. J., "An Optimal Design Procedure for Subband Vector Quantizers," submitted to *IEEE Trans. Commun.*, 1992.

Budge, S. E., Stockham, Jr., T. G., and Chabries, D. M., "Vector Quantization of Color Digital Images Within a Human Visual Model," *Proc. Int. Conf. on ASSP*, pp. 816–819, 1988.

Burt, P. J., and Adelson, E. H., "The Laplacian Pyramid as a Compact Image Code," *IEEE Trans. Commun.*, vol. COM-31, pp. 532–540, 1983.

Buzo, A., Gray, Jr., A. H., Gray, R. M., and Markel, J. D., "Speech Coding Based upon Vector Quantization," *IEEE Trans. Acous. Speech and Signal Process.*, ASSP-28, pp. 562–574, 1980.

Carlsson, S., and Reillo, C., "Contour Based Representation of the Displacement Field for Motion Compensated Image Coding," *Proc. IEEE Int. Conf. on ASSP*, pp. 161–164, 1986.

Chen, T., *Multidimensional Multirate Filters and Filter Banks: Theory, Design, and Implementation*, Dissertation, Electrical Engineering, Caltech, 1993.

Chen, T., and Vaidyanathan, P. P., "Recent Development in Multidimensional Multirate Systems," *IEEE Trans. Circuits and Systems for Video Technology*, vol. CSVT-3, no. 2, pp. 116–137, 1993.

Communications of the ACM, Special Issue on *Digital Multimedia Systems*, vol. 34, no. 4, 1991.

Crespo, J., and Serra, J. C., "Morphological Pyramids for Image Coding," *SPIE Visual Communications and Image Processing*, November 1993.

Fitch, J. P., Coyle, E. J., and Gallagher, Jr., N. C., "Median Filtering by Threshold Decomposition," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. ASSP-32, no. 6, pp. 1183–1188, 1984.

Foster, J., Gray, R. M., and Dunham, M., "Finite-State Vector Quantization of Waveform Coding," *IEEE Trans. Information Theory*, vol. IT-31, pp. 348–355, 1985.

Gallagher, N. C., and Wise, G. L., "A Theoretical Analysis of the Properties of Median Filters," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. ASSP-29, 1981.

Gersho, A., "Asymptotically Optimal Block Quantization," *IEEE Trans. Information Theory*, vol. IT-25, pp. 373-380, 1979.

Gersho, A., and Gray, R. M., *Vector Quantization and Signal Compression*, Norwell, Massachesetts: Kluwer Academic Publishers, 1991.

Ghanbari, M., "The Cross-Search Algorithm for Motion Estimation," *IEEE Trans. Commun.*, vol. COM-38, pp. 950–953, 1990.

Gharavi, H., and Tabatabai, A., "Subband Coding of Monochrome and Color Images," *IEEE Trans. Circuits and Systems*, vol. CAS-35, pp. 207–214, 1988.

Gray, R. M., "Vector Quantization," *IEEE ASSP Mag.*, vol. 1, pp. 4–29, 1984.

Hang, H. M., and Woods, J. W.,"Predictive Vector Quantization of Images," *IEEE Trans. Commun.*, vol. COM-33, pp. 1208–1219, 1985.

Huang, J., and Mersereau, R. M., "Contour-Based Hybrid Displacement Estimation for Image Sequence Compression," *Proc. IEEE Int. Conf. on ASSP*, vol. 5, pp. 433–436, 1993.

Huang, S. C., and Huang, Y. F., "Principal Component Vector Quantization for Abrupt Scene Changes," *Proc. Int. Symp. on CAS*, pp. 2280-2283, 1992.

Huang, T. S., Yang, G. J., and Tang, G. Y., "A Fast Two-Dimensional Median Filtering Algorithm," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. ASSP-27, no. 1, pp. 13–18, 1979.

IBM Journal of Research and Development, November 1988.

IEEE J. Selected Areas in Communcations, Special Issue on *Low Bit-Rate Coding of Moving Images*, vol. JSAC-5, no. 7, 1987.

IEEE J. Selected Areas in Communcations, Special Issue on *Packet Speech and Video*, vol. JSAC-7, no. 5, 1989.

IEEE J. Selected Areas in Communcations, Special Issue on *High-Definition Television and Digital Video Communications*, vol. JSAC-11, no. 1, 1993.

IEEE Trans. Circuits and Systems for Video Technology, Special Issue on *Packet Video*, vol. CSVT-3, no. 3, 1993.

IEEE Trans. Communications, Special Issue on *Picture Communication Systems*, vol. COM-29, no. 12, 1981.

Jain, J. R., and Jain, A. K., "Displacement Measurement and its Applications in Interframe Image Coding," *IEEE Trans. Commun.*, vol. COM-29, pp. 1799–1808, 1981.

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

Kappagantula, S., and Rao, K. R., "Motion Compensated Predictive Coding," in *Proc. Int. Tech. Symp. SPIE*, San Diego, CA, 1983.

Kim, C., Bruder, J., Smith, M. J. T., and Mersereau, R. M., "Subband Coding of Color Images using Finite State Vector Quantization," *Proc. Int. Conf. on ASSP*, pp. 753–756, April 1988.

Kim, D. S., and Lee, S. U., "Image Vector Quantizer Based on a Classification in the DCT Domain," *IEEE Trans. Commun.*, vol. COM-39, pp. 549–556, 1991.

Kim, J. T., Lee, H. J., and Choi, J. S., "Subband Coding Using Human Visual Characteristics for Image Signals," *IEEE Journal on Selected Areas in Communications*, vol. JSAC-11, pp. 59–64, January 1993.

Kim, Y. H., and Modestino, J. W., "Adaptive Entropy Coded Subband Coding of Images," *IEEE Trans. Image Processing*, vol. IP-1, pp. 31–49, 1992.

Koga, T., Iinuma, K., Hirano, A., Iijima, Y., and Ishiguro, T., "Motion Compensated Interframe Coding for Video Conferencing," in *Proc. Nat. Telecommunication Conf.*, New Orleans, LA, pp. G5.3.1–5.3.5, 1981.

Kunt, M., "Recent HDTV Systems," *Signal Processing VI: Theories and Applications*, pp. 83–89, Elsevier Science Publisher, 1992.

Kunt, M., Bénard, M., and Leonardi, R., "Recent Results in High-Compression Image Coding," *IEEE Tran. Circuit and Systems*, vol. CAS-34, pp. 1306–1336, Nov. 1987.

Kunt, M., Ikonomopoulos, A., and Kocher, M., "Second Generation Image Coding Techniques," *IEEE Tran. Communications*, vol. 73, pp. 549–574, April 1985.

Lee, M.-C., "Image Compression Using Generalized Multiscale Vector Quantization: A Product Code Vector Quantization Approach," *IEEE Data Compression Conf.*, March 1993a.

Lee, M.-C., "High Speed Video Compression Using Generalized Multiscale Motion Compensation with Logarithmic Quantization," *IEEE Data Compression Conf.*, March 1993b.

Lee, M.-C., "An Adaptive Perceptual Quantizer with Applications to Block Coding Systems," submitted to *IEEE Conf. on ASSP-94*, June 1993c.

Lee, M.-C., "An Ultrafast Video Compression Scheme Using Generalized Multirate Motion Compensation," *9th Computing in Aerospace Conference*, October 1993d.

Lee, M.-C., "A Low-Complexity Nonlinear Multiscale System Using the Median Filter for Image Compression," submitted to *IEEE Tran. Image Processing*, November 1993e.

Lee, M.-C., "A Median Filter Based Nonlinear Multiscale System for Image Compression," accepted *ISCAS'94*, July 1993f.

Le Gall, D., "MPEG: A Video Compression Standard for Multimedia Applications," *Communications of the ACM*, vol. 34, no. 4, pp. 47–58, 1991.

Lim, J. S., *Two-Dimensioanl Signal and Image Processing*, Englewood Cliffs, NJ: Prentice-Hall, 1990.

Linde, Y., Buzo, A., and Gray, R. M., "An Algorithm for Vector Quantizer Design," *IEEE Trans. Communications*, vol. COM-28, pp. 84–94, 1980.

Liou, M. L., "Overview of the $p \times 64$ Kbps Video Coding Standard," *Communications of the ACM*, vol. 34, no. 4, pp. 59–63, 1991.

Liu, B., and Zaccarin, A., "New Fast Algorithms for the Estimation of Block Motion Vectors," *IEEE Trans. Circuits and Systems for Video Technology*, vol. CSVT-3, no. 2., pp. 148–157, 1993.

Lowry, A., Hossain, S., and Millar, W., "Binary Search Trees for Vector Quantization," *ICASSP Proceedings*, pp. 51.8.1-51.8.4, 1987.

Mallat, S., "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation," *IEEE Trans. Pattern and Machine Intell.*, vol. PAMI-11, pp. 674–693, 1989a.

Mallat, S., "Multifrequency Channel Decompositions of Images and Wavelet Models," *IEEE Trans. Pattern and Machine Intell.*, vol. PAMI-37, pp. 2091–2110, 1989b.

Moorhead, II, R. J., Rajala, S. A., and Cook, L. W., "Image Sequence Compression Using a Pel-Recursive Motion-Compensation Technique," *IEEE J. Selected Areas in Communications*, vol. SAC-5, pp. 1100–1114, 1987.

Murkami, T., Asai, K., and Yamazaki, E., "Vector Quantization for Video Signals," *Electronics Letters*, 7: 1005–1006, November, 1982.

Nasrabadi, N. M., and King, R. A., "Image Coding Using Vector Quantization: A Review," *IEEE Trans. Communications*, vol. COM-36, pp. 957–971, 1988.

Netravali, A. N., and Limb, J. O., "Picture Coding: A Review," *Proc. of IEEE*, vol. 68, no. 3, pp. 366–406, 1980.

Netravali, A. N., and Mounts, F. W., "Ordering Techniques for Facsimile Coding: A Review," *Proc. IEEE*, vol. 68, pp. 796–806, July 1980.

Nguyen, T. Q., and Vaidyanathan, P. P., "Two-Channel Perfect-Reconstruction FIR QMF Structures Which Yield Linear-Phase Analysis and Synthesis Filter Banks," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-37, pp. 676–690, 1989.

Orchard, M., and Bouman, C., "Color Quantization of Images," *IEEE Trans. Signal Processing*, 39, pp. 2677–2690, 1991.

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

Petr, D. W., "32 kb/s ADPCM-DLQ Coding for Network Applications," *Proc. IEEE Globecom Conf.*, pp. A8.3.1–A8.3.5, December 1982.

Pitas, I., and Venetsanopoulos, A. N., *Nonlinear Digital Filters: Principles and Applications*, Norwell, MA: Kluwer Academic Publishers, 1990.

Preuss, D., "Comparison of Two-Dimensional Facsimile Coding Schemes," *Proc. Inter. Conf. on Communications*, pp. 7.12–7.16, June 1975.

Proc. of IEEE, Special Issue on *Digital Encoding of Graphics*, vol. 68, July 1980.

Puri, A., and Aravind, R., "Motion-Compensated Video Coding with Adaptive Perceptual Quantization," *IEEE Trans. Circuits and Systems for Video Technology*, vol. CSVT-1, pp. 351–361, December 1991.

Puri, A., Hang, H. M., and Schilling, D. L, "An Efficient Block-Matching Algorithm for Motion Compensated Coding," *Proc. IEEE Int. Conf. on ASSP*, pp. 25.4.1–25.4.4, 1987.

Ramamurthi, B., and Gersho, A., "Classified Vector Quantization," *IEEE Trans. Communications*, vol. COM-34, pp. 1105–1115, 1986.

Rao, V. V. B., and Rao, K. S., "A New Algorithm for Real-Time Median Filtering," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. ASSP-34, no. 6, pp. 1674–1675, 1986.

Raulin, J. M., Bonnerot, G., Jeandot, J. L., and Lacroix, R., "A 60 Channel PCM-ADPCM Converter," *IEEE Tran. Communications*, vol. COM-21, pp. 567–573, 1982.

Rioul, O., and Vetterli, M., "Wavelet and Signal Processing," *IEEE Signal Processing Magazine*, vol. 8, pp. 14–38, October 1991.

Sabin, M. J., and Gray, R. M., "Product Code Vector Quantizer for Waveform and Voice Coding," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, pp. 474–488, 1984.

Schalkoff, R. J., *Digital Image Processing and Computer Vision*, New York: John Wiley & Sons Inc., 1989.

Smith, M. J. T., and Eddins, S. L., "Analysis/Synthesis Techniques for Subband Image Coding," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. ASSP-38, no. 8, pp. 1446–1456, 1990.

Soman, A. K., and Vaidyanathan, P. P., "On Optimal Bit Allocation Strategies for Paraunitary Subband Coders," *Proc. of the 25th Annual Asilomar Conference on Signals, Systems and Computers*, pp. 741-745, 1991.

Soman, A. K., Vaidyanathan, P. P., and Nguyen, T. Q., "Linear-Phase Paraunitary Filter Banks: Theory, Factorizations and Applications," *IEEE Trans. Signal Processing*, December 1993.

Srinivasan, R., and Rao, K. R., "Predictive Coding Based on Efficient Motion Estimation," *Proc. IEEE Int. Conf. Communications*, Amsterdam, pp. 521–526, 1984.

Stockham, T. G., Jr., "Image Processing in the Context of a Visual Model," *Proc. IEEE*, vol. 60, no. 7, pp. 828–842, 1972.

Vaidyanathan, P. P., and Hoang, P.-Q., "Lattice Structure for Optimal Design and Robust Implementation of Two-Channel Perfect-Reconstruction," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-36, pp. 81–94, 1988.

Vaidyanathan, P. P., "Multirate Digital Filters, Filter Banks, Polyphase Networks, and Applications: A Tutorial," *Proc. IEEE*, vol. 78, pp. 56–93, 1990.

Vaidyanathan, P. P., *Multirate Systems and Filter Banks*, Englewood Cliffs, NJ: Prentice-Hall, 1993.

Vaisey, J., and Gersho, A., "Image Compression with Variable Block Size Segmentation," *IEEE Trans. Signal Processing*, vol. SP-40, no. 8, pp. 2040–2060, 1992.

Vetterli, M., "Multidimensional Subband Coding: Some Theory and Algorithms," *Signal Processing*, vol. 6, no. 2, pp. 97–112, 1984.

Vetterli, M., and Uz, K. M., "Multiresolution Coding Techniques for Digital Television: A Review," *Multidimensional Systems and Signal Processing*, vol. 3, pp. 161–187, 1992.

Wallace, G. K., "The JPEG Still Picture Compression Standard," *Communication of the ACM*, vol. 34, pp. 31–44, April 1991.

Ward, J., "Hierarchical Grouping to Optimize an Objective Function," *J. Amer. Stat. Assoc.*, vol. 37, pp. 236–244, 1963.

Weber, D. R., "An Adaptive Run-Length Coding Algorithm," *Proc. Int. Conf. on Communication*, pp. 7.4–7.7, June 1975.

Westerink, P. H., Boekee, D. E., Biemond, J., and Woods, J. W., "Subband Coding of Images Using Vector Quantization," *IEEE Trans. Communications*, vol. COM-36, pp. 713–719, 1988.

Wickerhauser, M. V., "High-Resolution Still Picture Compression," *Digital Signal Processing 2*, pp. 204–226, 1992.

Witten, I. H., Neal, R. M., and Cleary, J. G., "Arithmetic Coding for Data Compression," *Communications of the ACM*, vol. 30, pp. 520–540, June 1987.

Wong, Y.-F., "Ultrafast Principal Component Vector Quantization," *Proc. IEEE Data Compression Conf.*, 1993a.

Wong, Y.-F., "Clustering Data by Melting," *Neural Computation*, vol. 5, No. 1, 1993b.

Wong, Y.-F., "A Clustering Filter for Scale-Space Filtering, Image Restoration and Nonparametric Regression," *IEEE Conference on Computer Vision and Pattern Recognition*, May, 1993c.

Wong, Y.-F., and Lee, M.-C., "Image Compression by a Nonlinear Multiresolution System," *Proc. IEEE Workshop on Visual Signal Processing and Communications*, September 1993a.

Wong, Y.-F., and Lee, M.-C., "Using a Nonlinear Multiresolution System for Image Compression," submitted to *IEEE Trans. Circuits and Systems for Video Technology*, May 1993b.

Woods, J., *Subband Image Coding*, Norwell, MA: Kluwer Academic Publishers, 1991.

Woods, J. W., and O'Neil, S. D., "Subband Coding of Images," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. ASSP-34, no. 5, pp. 1278–1288, 1986.

Wu, X., "Vector Quantizer Design by Constrained Global Optimization," *Proc. IEEE Data Compression Conf.*, pp. 132–141, 1992.

Wu, X., and Zhang, K., "A Better Tree-Structured Vector Quantizer," *Proc. IEEE Data Compression Conf.*, pp. 392-401, 1991.

Xie, Z., and Stockham, T. G., Jr., "Previsualized Image Vector Quantization with Optimized Pre- and Postprocessors," *IEEE Trans. Communications*, vol. COM-39, no. 11, pp. 1662–1671, 1991.

Yasuda, Y., "Overview of Digital Facsimile Coding Techniques in Japan," *Proc. IEEE*, vol. 68, pp. 830–845, July 1980.

Zador, P. C., "Asymptotic Quantization of Continuous Random Variables," *IEEE Trans. Information Theory*, vol. IT-28, pp. 139-159, 1982.

Zafar, S., Zhang, Y.-Q., and Jabbari, B., "Multiscale Video Representation Using Multiresolution Motion Compensation and Wavelets Decomposition," *IEEE J. Selected Areas in Communications*, vol. JSAC-11, no. 1, pp. 24–35, 1993.

Zhang, Y.-Q., and Zafar, S., "Motion-Compensated Wavelet Transform Coding for Color Video Compression," *IEEE Trans. Circuit and Systems for Video Technology*, vol. CSVT-2, pp. 285–296, 1992.

Selected References by Topic

Properties of HVS

Baseri and Mathews, 1992
Bradley et al., 1992
Budge et al., 1988
Kim et al., 1993
Lim, 1990
Schalkoff, 1989
Stockham, 1972
Xie and Stockham, 1991

Median Filters

Arce and McLoughlin, 1987
Arce and Stevenson, 1987
Fitch et al., 1984
Gallagher and Wise, 1981
Pitas and Venetsanopoulos, 1988

Nonlinear Multiscale Techniques

Burt and Adelson, 1983
Crespo and Serra, 1993
Lee, 1993f
Vetterli and Uz, 1992
Wong and Lee, 1993a
Wong and Lee, 1993b

DCT Based Coding Schemes

Jayant and Noll, 1984
Le Gall, 1991
Liou, 1991
Netravali and Limb, 1980
Pennebaker and Mitchell, 1993
Wallace, 1991
Wickerhauser, 1992

Wavelets and Filter Banks

Chen, 1993
Mallat, 1989a
Mallat, 1989b
Rioul and Vetterli, 1991
Vaidyanathan, 1990
Vaidyanathan, 1993
Vetterli and Uz, 1992

Subband Image and Video Coding

Antonioni et al., 1992
Bradley et al., 1992
Chen and Vaidyanathan, 1993
Gharavi and Tabatabai, 1988
Kim et al., 1988
Kim and Lee, 1991
Smith and Eddins, 1990
Vetterli, 1984
Westerink et al., 1988
Wickerhauser, 1992
Woods, 1991
Woods and O'Neil, 1986
Zafar et al., 1993

Vector Quantization

Foster et al., 1985
Gersho, 1979
Gersho and Gray, 1991
Gray, 1984
Hang and Woods, 1985

Lee, 1993a
Linde et al., 1980
Lowry et al., 1987
Murkami et al., 1982
Nasrabadi and King, 1988
Ramamurthi and Gersho, 1986
Sabin and Gray, 1984
Wu and Zhang, 1991
Xie and Stockham, 1991

Motion Estimation/Compensation

Biemond et al., 1987
Carlsson and Reillo, 1986
Ghanbari, 1990
Huang and Mersereau, 1993
Jain and Jain, 1981
Kappagantula and Rao, 1983
Koga et al., 1981
Lee, 1993b
Lee, 1993d
Liu and Zaccarin, 1993
Moorhead II et al., 1987
Puri et al., 1987
Zafar et al., 1993

Packet Video

IEEE CSVT, 1993
IEEE JSAC, 1989

Second-Generation Coding Schemes

Kunt et al., 1985
Kunt et al., 1987

Books and Papers of Tutorial Value

Communications of the ACM, 1991
Gersho and Gray, 1991
Gray, 1984
IEEE Commun., 1981
IEEE JSAC, 1987
IEEE JSAC, 1993
Jayant and Noll, 1984
Kunt, 1992
Rioul and Vetterli, 1991
Vaidyanathan, 1993
Vetterli and Uz, 1992
Wickerhauser, 1992
Woods, 1991