THE HONG KONG
POLYTECHNIC UNIVERSITY
香港理工大學
Pao Yue-kong Library
包玉剛圖書館

Copyright Undertaking

This thesis is protected by copyright, with all rights reserved.

By reading and using the thesis, the reader understands and agrees to the following terms:

1. The reader will abide by the rules and legal ordinances governing copyright regarding the use of the thesis.
2. The reader will use the thesis for the purpose of research or private study only and not for distribution or further reproduction or any other purpose.
3. The reader agrees to indemnify and hold the University harmless from and against any loss, damage, cost, liability or expenses arising from copyright infringement or unauthorized usage.

If you have reasons to believe that any materials in this thesis are deemed not suitable to be distributed in this form, or a copyright owner having difficulty with the material being included in our database, please contact lbsys@polyu.edu.hk providing details. The Library will look into your claim and consider taking remedial action upon receipt of the written requests.

Pao Yue-kong Library, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

http://www.lib.polyu.edu.hk

Content-based and Temporal-scalable Video Coding

by

HO Kai-Hong

A dissertation submitted in partial fulfillment of the requirements for the degree of

Master of Philosophy

Department of Electronic and Information Engineering
The Hong Kong Polytechnic University

2002

Pao Yue-Kong Library
PolyU • Hong Kong

DEDICATION

To my parents.

ABSTRACT of thesis entitled
'Content-based and temporal-scalable video coding'
submitted by HO Kai-Hong for the degree of Master of Philosophy
at The Hong Kong Polytechnic University in 2002.

In the design of video coding systems, compression efficiency and scalability are two of the most important topics under investigation. While channel bandwidth is still the most precious resource for many data networks, we require video coding algorithms that allow very high compression rate, yet retain reasonably good visual quality. To achieve this, the concept of content-based video coding is proposed recently to address the need of considering the features of video content to improve compression efficiency. On the other hand, scalability is another important aspect for video coding as it allows adaptation of video quality to different constraints incurred in the video transmission or retrieval processes.

In this thesis, we firstly investigate the content-based video coding algorithms for low bit rate networks. The process of content-based video coding is divided into three parts: (i) object segmentation; (ii) feature extraction; (iii) adaptation to video coding systems. To demonstrate the approach, we apply the idea to the design of a content-based H.263 video codec for real-time coding of road traffic videos. Based on the proposed approach, moving objects (i.e. cars) are first segmented and classified based on their velocity. They are then coded in different frame rates accordingly. As compared with the conventional H.263 encoder using for the same application, the proposed system has a 20% increase in compression rate with negligible visual distortion. The proposed system fully complies with the ITU H.263 standard hence the encoded bit stream is completely comprehensible to the conventional H.263 decoder.

As for the scalable video coding, a new temporal-scalable video codec is proposed in this thesis to provide compressed videos at different frame rates. The proposed codec is developed based on the interpolating wavelet transform. It shares the same advantage of the traditional temporal subband (TSB) approach in that its structure is very simple since it does not require the complicated motion compensation process. It outperforms substantially the TSB approach in generating lower frame rate videos.

ACKNOWLEDGEMENTS

I would like to express sincere gratitude to various bodies from The Hong Kong Polytechnic University, where I have the opportunity to work with. I am very thankful to my supervisor Dr. Daniel Pak Kong Lun. He carefully guided me throughout the pursuit of my Master's degree at Polytechnic University, his freedom allowed me to have second though of research problems and his ideas and suggestions have also been invaluable to this thesis.

Many thanks to my colleagues, Dr. T.C. Hsung, Mr. C. L. Chan, Mr. W. K. Cheuk for their insightful suggestions and assistance at every stage of my work. I would also like to thank Mr. S. M. Cheung, Mr. Steven Leung, Mr. S. K. Wan, Ms. Y. F. Ho, Mr. W. L. Hui, Mr. K. C. Lam, Mr. Dylan Lu, Mr. Roscoe Cheung, Mr. Leo Liu, and those who have worked with me in the last two years. The countless discussion with them have proved to be fruitful and inspiring.

I also have to thank all members of staff of the department of Electronic and Information Engineering in the Hong Kong Polytechnic University, especially, Prof. W. C. Siu, Dr. Kenneth Lam, Dr. Christie Chan and the clerical staff in the General Office. They have created a supportive environment for me to work.

It is my pleasure to acknowledge to the Hong Kong Polytechnic University research grant for providing financial support to me which made my work and thesis possible.

Last in this list but first in my heart, I must express my heartfelt gratitude to my family for the support and encouragement. Without them, this study would not have the chance to be completed.

STATEMENT OF ORIGINALITY

The following contributions reported in this thesis are claimed to be original.

1. An improved real-time video segmentation algorithm (Chapter 3, Section 3.2.1).

   In this algorithm, change detection mask (CDM) is found by applying a global threshold to luminance difference between two successive frames and the results are mapped into 8x8 data blocks. This segmentation algorithm can effectively segment moving objects from video in real time and can be adopted to current block-based video codec.

2. A content-based scalable video coding algorithm (Chapter 3, Section 3.3).

   A content-based scalable H.263 video coding system that is suitable for low bit rate video applications is suggested. It has been shown that under a low bit rate condition, the proposed system can provide a consistent improvement in terms of PSNR when comparing with the conventional H.263 codec. Besides, the encoded video sequence can be decoded with conventional H.263 decoder which is different from other proprietary codec that requires a tailor-made decoder.

3. A novel approach on feature extraction for road traffic video segmentation (Chapter 4, Section 4.4.1 and 4.4.2).

   As mentioned in the part of real-time segmentation, we segment out the moving objects from video sequence. We apply this algorithm in road traffic video, and those moving objects are further analyzed and classified as high or low activity by assessing the regularity of their activities using the proposed correlation approach or zero crossing density detection approach. This feature extraction can further improve the segmentation by introducing one more object plane.

4. The content-based scalable coding for road traffic surveillance system (Chapter 4, Section 4.5.1).

   We further apply the abovementioned content-based H.263 video coding scheme to road traffic monitoring. Rather than only differentiating moving objects (i.e. cars in this application) from static background, we improve the coding scheme by further differentiating fast moving objects from slow moving objects and assigning different resource for their coding. It has been shown that the bit rate can be reduced by more than 20%. The encoded video sequence can be decoded with conventional H.263 decoder, which is different from other proprietary codec that requires a tailor-made decoder. Besides road traffic monitoring, the proposed system can also be applied to other video surveillance systems with fixed camera setting.

5. Temporally scalable video coding using interpolating wavelet transform (Chapter 5, Section 5.2 and 5.4).

   We propose a new temporally scalable video coding algorithm based on the interpolating wavelet transform. With the proposed approach, the input video frames are first applied to an interpolating wavelet transform which generates video frames with reduced temporal redundancy in its high pass branch and original video frames at lower rate in its low pass branch. The proposed video codec shares the same advantage of the temporal subband coding in that it is very simple in nature since it does not require the complicated motion compensation process. It outperforms substantially the temporal subband coding in generating lower frame rate videos.

6. The reversible rounding scheme for temporal scalability (Chapter 5, Section 5.3).

   In this scheme, we suggest a reversible rounding method to convert the floating point coefficients given by the interpolating wavelet transform into integers without loss of resolution. It further improves the compression efficiency by not encoding the decimal part of the coefficients which can be regenerated at the decoder side.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

AUTHOR'S PUBLICATIONS

International Journal Papers

1. Kai-Hong Ho and Daniel, Pak-Kong Lun, "Content-based Scalable H.263 Video Coding for Road Traffic Monitoring," IEEE Transactions on Multimedia, (submitted).

2. Kai-Hong Ho and Daniel, Pak-Kong Lun, "On Temporally Scalable Video Coding Using Interpolating Wavelet Transform," IEEE Transactions on Circuits and Systems for Video Technology, (submitted).

International Conference Papers

3. Kai-Hong Ho and Daniel Pak-Kong Lun, "Content-Based Scalable H.263 Video Coding Scheme for Road Traffic Monitoring," Proceeding of International Workshop on Multimedia Data Storage, Retrieval, Integration and Applications (MMWS'2000), Hong Kong, 13 – 15 January, 2000, pages 163 – 170, (2000).

4. Kai-Hong Ho and Daniel Pak-Kong Lun, "Road Traffic Monitoring System using Content-based Scalable H.263 Video Coding Scheme," Proceeding of IEEE Pacific-Rim Conference on Multimedia (IEEE-PCM 2000), Australia, 13 – 15 December, 2000, pages 257 – 260 (2000).

5. Kai-Hong Ho and Daniel Pak-Kong Lun, "Content-based scalable H.263 video coding for road traffic monitoring based on regularity of video content," Proceeding of IEEE International Symposium on Intelligent Multimedia, Video and Speech Processing (ISIMP 2001), Hong Kong, 2 – 4 May, 2001, pages 324 – 327 (2001).

6. Kai-Hong Ho and Daniel Pak-Kong Lun, "Efficient Wavelet-based Temporally Scalable Video Coding," Submitted to 6th International Conference on Signal Processing (ICSP'02), Beijing, China, 26 – 30 August, 2001, pages 881 – 884.

7. L. K. Wong, S. H. Ling, Frank Leung, Y. S. Lee, S. H. Wong, T.H. Lee, H. K. Lam, Kai-Hong Ho, Daniel P. K. Lun and T. C. Hsung, "An Intelligent Home", Proceedings, CIDAM Workshop on Service Automation and Robotics, June 2000, Hong Kong, pages 111 – 119.

Chapter 1

Introduction

1.1 Needs for compression

Video compression is a process that enables the effective bit rate of video to be reduced for transmission or storage. It is well known that, for current digital systems, video information is not represented in the most efficient way. Much information can be reduced without affecting the visual quality. By video compression, we can better utilize the scarce resource, such as bandwidth or memory, in video transmission and storage.

In general, the information of a video sequence is highly correlated. There are two categories of correlation, namely, spatial and temporal. The color of an image in a local region typically does not abruptly change; hence there is spatial correlation. Unless there is a scene change in the video, there is not much change between two consecutive video frames, and this is called temporal correlation.

The existence of correlation implies that we can easily estimate one information from the knowledge of another. This indicates the existence of redundancy in the information. Much research work has been conducted to reduce such spatial and temporal

redundancies in video. The technology for video compression has matured to a stage that various compression standards have been promulgated to enable interoperability of different equipment manufacturers.

1.2 Video compression standards

Recently, many new digital video applications are emerging. Ranging from low bit rate video systems such as videoconferencing, video streaming, wireless video and video surveillance systems to high bit rate video systems such as "video-on-demand" and high definition television (HDTV) systems, they have been substituting their traditional analog counterparts. They also enable new services that were not previously possible such as medical imaging and telemedicine.

With the rapid growth of digital video applications, powerful data compression techniques with different quality criteria are needed. However, practical applications require that all users who wish to interchange their compressed video must use exactly the same compression algorithm. Also, further sophisticated algorithms will benefit from the development of special hardware. All these express the need for standards to allow the orderly growth of markets which utilize video compression technology.

Driven by these needs, there has been a strong effort to develop international standards for motion video compression algorithms, underway for several years in the International Standards Organisation (ISO) and the International Electrotechnical Commission (IEC). It is the Motion Pictures Expert Group (MPEG) which considers algorithms for motion video compression. Subsequently, a number of standards have been

defined for the compression of visual information. Table 1.1 summarizes common video compression standards [1].

Standard	Standard-ization body	Main target bit rate	Main compression technologies	Main target applications
JPEG2000	ISO/IEC	Compression ratios 2-50	– Wavelet – Perceptual quantization – Visual frequency weighting – Arithmetic coding – Region of interest coding – Error resilience coding	– Internet imaging – Digital photography – Image and video editing – Printing – Medical imaging – Mobile applications – Color fax – Satellite imaging
MPEG-1	ISO/IEC	1.5Mbit/s	– DCT – Perceptual quantization – Zig-zag reordering – Predictive motion compensation – Bi-directional motion compensation – Half-sample accuracy motion estimation – Huffman coding – Arithmetic coding	– Storage on CD-ROM – Consumer video
MPEG-2	ISO/IEC	1.5-35Mbit/s	– DCT – Perceptual quantization – Adaptive quantization – Zig-zag reordering – Predictive motion compensation – Bi-directional motion compensation – Frame/field based motion estimation – Half-sample accuracy motion estimation – Spatial scalability – Temporal scalability – Quality scalability – Huffman coding – Arithmetic coding – Error resilient coding	– Digital TV – Digital HDTV – High quality video – Satellite TV – Cable TV – Terrestrial broadcast – Video editing – Video storage
MPEG-4	ISO/IEC	8Kbit/s – 35Mbit/s	– DCT – Wavelet – Perceptual quantization	– Internet – Interactive video – Visual editing

			– Adaptive quantization – Zig-zag reordering – Zero-tree reordering – Predictive motion compensation – Bi-directional motion compensation – Frame/field based motion estimation – Half-sample accuracy motion estimation – Advanced motion estimation – Overlapping motion estimation – Spatial scalability – Temporal scalability – Quality scalability – View dependent scalability – Bitmap shape coding – Sprite coding – Face animation – Dynamic mesh coding – Huffman coding – Arithmetic coding – Error resilient coding	– Content manipulation – Consumer video – Professional video – 2D/3D computer graphics – Mobile
H.261	ITU-T	P x 64 Kbit/s	– DCT – Adaptive quantization – Zig-zag reordering – Predictive motion compensation – Integer-sample accuracy motion estimation – Huffman coding – Error resilient coding	– ISDN video-conferencing
H.263	ITU-T	8 Kbit/s-1.5Mbit/s	– DCT – Adaptive quantization – Zig-zag reordering – Predictive motion compensation – Bi-directional motion compensation – Half-sample accuracy motion estimation – Advanced motion estimation – Overlapping motion estimation – Huffman coding – Arithmetic coding – Error resilient coding	– POTS video-telephony – Desktop video telephony – Mobile video telephony

Table 1.1. Summary of present standards for video coding, adopted from [1].

MPEG-1 [2] operates at bit rates around 1.5Mbit/s. Its targets are to enable audio and video to be compressed and stored onto a compact disc as well as transmitted through a network with narrow bandwidths such as, integrated services digital network (ISDN). MPEG-2 [3] operates at bit rates of up to about 35Mbit/s and provides high-quality video applications, such as HDTV and digital video disc, at the expense of more complex processing compared with MPEG-1. Besides, it has defined several new profiles such as spatial scalability and temporal scalability.

MPEG-4 is a recent standard made for interactive video on CD-ROM and Digital Television. It was finalized in October 1998 and became an International Standard in 1999. One of the targets of MPEG-4 is to provide a unified coding and authoring environment for integrating synthetic data with natural video data. It tries to provide interactive video applications including interactive multimedia broadcast and mobile communications.

To encode videos with a bit rate less than 64Kbit/s, H.261 [4] and H.263 [5] are commonly used for low bit rate video applications. H.263 provides more advanced compression techniques than H.261 and is used to transmit video in mobile channels with a bit rate as low as 9.6Kbit/s. H.263 Version 2, also known as H.263+, is an extension of H.263 that was officially approved as a standard in January 1998 [6]. This extension provides better compression performance, scalable bit streams, network packetization support, custom picture size and clock frequency support. A detailed tutorial of H.263+ can be found in [7]. Meanwhile, H.263++ is currently under development by the ITU-T.

Motion JPEG2000 (MJP2) [10] is a new video coding system based on the image coding standard JPEG-2000 [11]. The target market for MJP2 is very large, ranging from

Internet video to medical imaging. Different from current video coding schemes, MJP2 provides a new intra-based coding system and no motion prediction is employed. It is suitable to applications where high quality video is required.

1.3 Present work

New video applications are emerging from time-to-time. While channel bandwidth is still the most precious resource for many data networks, to meet the requirements of new applications, we require more powerful video coding algorithms that allow very high compression rate, yet retain reasonably good visual quality. The objective of this study is to investigate the advanced video compression methods. In particular, we have focused on two important techniques in video coding, namely, content-based technique and temporal-scalable technique for low bit rate video coding. The concept of content-based video coding has been suggested for long and particularly in the MPEG-4 standard [12]. It suggests that the knowledge of the image content can greatly benefit the coding performance. Nevertheless, a sophisticated implementation of this concept is yet to be investigated. As for the scalable video coding, it suggests that a video codec should be able to adapt to the constraint in the environment to adjust the compression ratio. It is noticed that new video coding standards often incorporate such useful feature as part of their specifications. There are several kinds of scalable video coding technique in the literature, namely resolution scalability, SNR scalability and temporal scalability, with the latter one be the easiest for implementation in practical systems. Furthermore, temporal scalability in general can work with other kind of scalability techniques without changing the bit stream format. They are the reasons why the study of temporal scalability has attracted much attention recently. Although there are many temporally scalable video

coding methods available, the advent of new signal processing techniques incurs us to have a second thought of the problem. More specifically, the present work can be separated into the following three different parts.

1.3.1 Real-time video segmentation for content-based coding

For sending video data through very low bit rate networks in real-time, video codec with high compression rate is the pre-requisite. Although there are many standard video codecs, such as H.263, have been suggested for this kind of applications, it is generally believed that their compression efficiency can be further improved if the content-based scalable video coding technique can be applied. The process of content-based video coding can be divided into three parts: (i) object segmentation; (ii) feature extraction; (iii) adaptation to video coding systems. In this thesis, a content-based H.263 video codec that is suitable for general video applications is proposed. In the proposed coding scheme, object motion is used as a cue for object segmentation. Moving objects are then differentiated from the static background and are encoded separately. To summarize, the major contributions of this research work are (i) the introduction of a real-time video segmentation unit; (ii) the modification of the control unit of H.263 that the coding of the static background blocks are automatically skipped and (iii) the encoded sequence is completely comprehensible to the conventional H.263 decoder. As compared with the conventional H.263 codec, consistent improvement in terms of peak signal to noise ratio (PSNR) is obtained in the decoded video with the same bit rate.

1.3.2 Application of content-based H.263 video coding in road traffic monitoring

We further apply the abovementioned content-based H.263 video coding scheme to road traffic monitoring [9]. Rather than only differentiating moving objects (i.e. cars in this application) from static background, we improve the coding scheme by further differentiating fast moving objects from slow moving objects and assigning different resource for their coding. More specifically, input road traffic video is first processed by a real-time segmentation [8] unit that the image blocks with moving objects are separated from those with steady background. Those image blocks with moving objects are further analyzed and classified as high or low activity by assessing the regularity of their activities using the proposed correlation approach or zero crossing density detection approach. All image blocks are then coded in 3 different frame rates hence the temporal redundancy of the video is greatly reduced. The frame rate control is achieved externally by adaptively adjusting a threshold value of the segmentation unit. This avoids a drastic modification of the internal control mechanism of the H.263 encoder for implementing the complicated control logic. Besides the reduction of temporal redundancy, a psychovisual thresholding unit is introduced into the H.263 encoder to further reduce the spatial redundancy of the image blocks. It is possible since human beings are more sensitive to slow moving objects. As compared with the conventional H.263 codec, the proposed system improves the compression rate by more than 20% with negligible visual distortion.

1.3.3 Temporally scalable video coding using interpolating wavelet transform

As it is mentioned above, the study on temporally scalable video coding methods has attracted much attention. Traditional approaches implement temporal scalability by either introducing extra reference frames to the motion compensated (MCP) video coding algorithms or simply switching to the temporal subband (TSB) video coding approaches.

While the MCP approaches introduce extra complexity to the already complicated motion compensation process, the TSB approach may give a substantially degraded performance particularly for lower frame rate videos. In this thesis, we propose a new temporally scalable video coding algorithm based on the interpolating wavelet transform. With the proposed approach, the input video frames are first applied to an interpolating wavelet transform which generates video frames with reduced temporal redundancy in its high pass branch and original video frames at lower rate in its low pass branch. We further propose the reversible rounding method to convert the floating point coefficients given by the interpolating wavelet transform into integers without loss of resolution. The proposed video codec shares the same advantage of the TSB approach in that it is very simple in nature since it does not require the complicated motion compensation process. It outperforms substantially the TSB approach in generating lower frame rate videos.

1.4 Organization of the thesis

This thesis is organized as follows. In Chapter 2, we give a review on the existing video coding schemes. Their strengths and weakness are discussed. In Chapter 3, a content-based H.263 video coding scheme is proposed and in Chapter 4, this coding scheme is further extended and applied to road traffic monitoring. We focus on temporally scalable video coding in Chapter 5. We illustrate how we can make use of the interpolating wavelet transform to achieve bit stream scalability in temporal domain. Finally, a summary of the work done is given in Chapter 6, where future extensions of the present work are also discussed.

Chapter 2

Review on video coding schemes

2.1 Introduction

In this chapter, some of the advanced video coding schemes are reviewed. First we outline some commonly used compression techniques on pixel based video coding. Then, some of the recent advances in video compression schemes, also known as second generation video coding, are elaborated in detail. They are (i) content-based video coding and (ii) temporal scalability techniques.

2.2 Fundamentals of pixels-based techniques

2.2.1 Introduction

This section presents the main concepts on which pixel-based image and video compression techniques are based. A summary of the compression techniques is shown in Figure 2.1. The assumptions are that the input is always a digitized signal in color components and the output of the compression process is a bit stream. Let us consider each technique briefly.

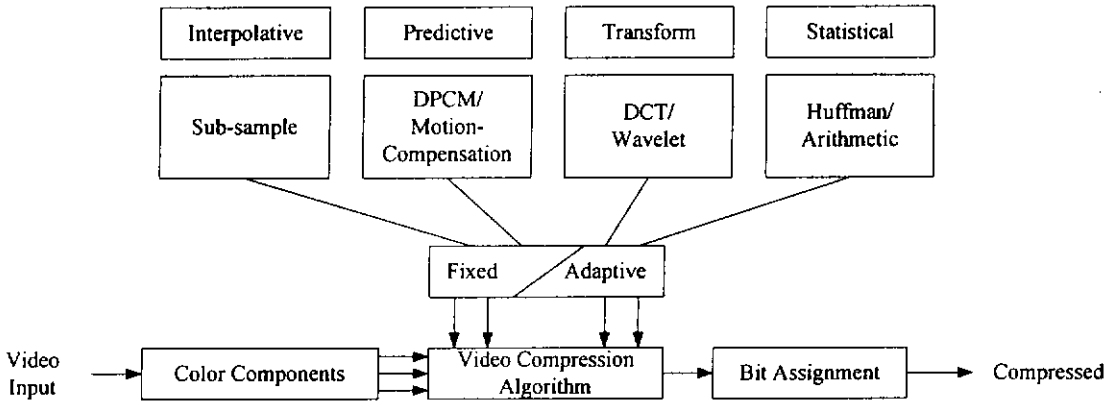


Fig. 2.1. Basic compression techniques.

2.2.2 Subsampling

Subsampling reduces the amount of data of discrete images or video signals by changing the sampling structure into another one with a reduced sampling rate (either in spatial domain, temporal domain or both) [13].

For video signals, subsampling can be classified into progressive and interlaced. In progressive mode, a single instance in time is enough to cover the complete spatial area. For interlaced videos, line interlaced instances in time are needed to cover the complete image area. The complete images, or frames, are constructed by the partial images called field.

2.2.3 Spatial transform

For a video signal, statistical redundancy can be found in both spatial and temporal domain. In spatial domain, there are two main approaches for eliminating the statistical redundancy in the signal: the predictive approach and the transformational approach [14]. The predictive approach is carried out by directly computing the signal elements from its neighborhood elements as their values are highly correlated. A transform is a process that converts data into an alternate form which is more convenient for some particular purpose. Transforms are usually designed to be reversible.

Differential Pulse Code Modulation (DPCM) is a kind of predictive coding. It operates at the pixel level and sends only the difference between successive pixels. Since there is likely to be very little difference between adjacent pixels, we can encode the value into smaller data widths by using DPCM. However, this technique suffers from slope-overload which causes smearing at high contrast edges in an image. ADPCM (Adaptive DPCM) however, tries to reduce the slope-overload by using different step sizes for difference values.

Transform coding has been studied extensively for many years and has become a very popular compression method for still-image coding and video coding. The purpose of transform is to de-correlate the image content and to encode the transform coefficients rather than the original pixels' values. It tries to make the transform coefficients as small as possible to minimize the statistical dependency between transform coefficients so that less bit is needed to encode those coefficients.

Upon many possible alternatives, the Discrete Cosine Transform (DCT) is especially important for still image and video compression [15]. It is due to the fact that

this transform has high decorrelation performance and the availability of fast algorithms for its implementation [16], [17] and [18]. The DCT is performed on an image block of horizontally and vertically adjacent pixels (typically an 8 by 8 block of pixels). The outputs represent amplitudes of two-dimensional spatial frequency components. These are called DCT coefficients. The coefficient for zero spatial frequency is called the DC coefficient and it is the average value of all the pixels in the block. The rest of the coefficients represent progressively higher horizontal and vertical spatial frequencies in the block.

For the traditional DCT-based coding methods, each of the DCT coefficients is completely encoded before the coding of the next coefficient. However, it often introduces the rate-control problem since if for any reason that the encoding process terminates on the way, only some of the transform coefficients are obtained. Hence, several embedded DCT-based coding schemes have been proposed that share many of the ideas used in wavelet-based coders [19], [20] and [21]. The embedded coding scheme firstly quantizes the coefficient into a certain number of bits and orders that according to their significance. The coding order is consistent with the importance of each bit so that the encoder and decoder can stop at any time. The embedding property is essential for progressive image transmission. It also greatly simplifies the rate control problem and allows an unequal error protection for robust image transmission.

The wavelet transform has recently emerged as a promising technique for image processing due to its flexibility in representing nonstationary image signals, and its ability in adapting to human visual characteristics. It has superior performance when compared with the DCT for image compression [22] and [23]. Basically, the wavelet transform

divides a signal into number of segments, each corresponding to different frequency bands. The wavelet representation provides a multiresolution expression [23] of a signal with localization in both time and frequency domain, hence the transformed signals can be processed much easier than the original signals. In image applications, wavelet transform decomposes an image signal into a set of subband as shown in Figure 2.2, so that bit allocation in each band can be performed according to human visual characteristics.

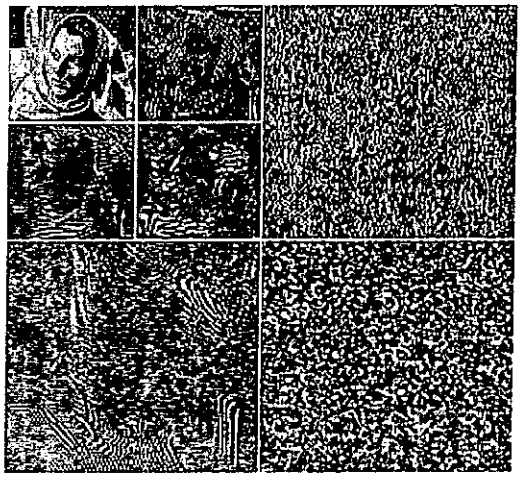


Fig. 2.2. Wavelet Decomposition of *Woman* using 9-3 tap biorthogonal filter.

2.2.4 Spatial-temporal transform

Apart from removing spatial redundancy, video compression also relies on removing temporal redundancy from the source signal. For consecutive video frames, it is noticed that the motion change of a rigid body for each frame is very small; the context of consecutive frames is highly correlated. The most popular as well as efficient approach to reduce such temporal correlation between consecutive frames of a video signal is by means of motion compensated prediction (MCP). MCP makes use of the motion field of a

video signal at a given instance to predict its following instance. This process is known as motion estimation and is similar in principle to that of predictive coding.

The MCP technique is employed in video coding standards such as H.261/H.263 and MPEG. The most popular MCP techniques are those making use of prediction from the past frames (predictive) or past and future frames (bi-directional), and those using overlapping motion compensation [5] and [12]. The implementation of MCP for the standards is by means of the block matching algorithm (BMA). The most straightforward BMA is the full search algorithm, which exhaustively searches for the best matched block within a search area in the previous frame to get the optimal motion vector. However the computational complexity of this method is very high. Thus, efficient algorithms such as the four step search algorithm [24], new three step search algorithm [25], three step search algorithm [26], the conjugate direction search algorithm [27], the 2-D logarithm search [28] and the cross search algorithm [29] are proposed for fast computation of MCP.

After the estimation process, the predicted error, or displaced frame difference (DFD) for each block is found. And this DFD is then transformed spatially using techniques such as DCT or wavelet transform.

2.2.5 *Quantization*

Quantization refers to a process of approximating the continuous set of values in the image data with a finite set of values. The input to a quantizer is the original data, and the output is always one among a finite number of levels. The quantizer is a function whose set of output values are discrete, and usually finite. Obviously, this is a process of

approximation, and a good quantizer is one which represents the original signal with minimum loss or distortion.

Quantization can be classified into two classes: scalar quantization and vector quantization. For a scalar quantizer, each input data is treated separately in producing the output symbol. The quantizer can be specified by its input partitions and output levels (also called reproduction points). If the input range is divided into levels of equal spacing, then the quantizer is termed as a uniform quantizer, otherwise it is termed as a non-uniform quantizer. A uniform quantizer can be easily specified by its lower bound and the step size. Also, implementing a uniform quantizer is easier than a non-uniform quantizer. A typical uniform quantization is shown in Figure 2.3. If the input x falls between $n*r$ and $(n+1)*r$ where n is an integer and r is quantization step size, then the quantizer outputs the symbol x'. The quantization error can be defined as $(x-x')$ and it is used as a measure of the optimality of the quantizer and dequantizer.

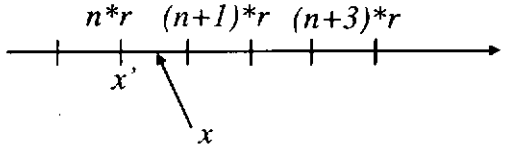


Fig. 2.3. Example of uniform scalar quantization.

Another type of quantization is vector quantization (VQ). VQ process involves three phases namely, training, encoding and decoding. The training phase is a codebook design procedure which tries to find the best set of representatives (codewords) from a large set of training vectors. At the encoding phase, input data are clubbed together in

groups called vectors, and processed to give the output. Suppose we have several data at the same time with an input vector of

$$\mathbf{s} = (s_1, \ldots, s_N) \tag{2.1}$$

with N values. These values can be, for example, pixels lying next to each other. The input vector is then matched to a set of codewords $\mathbf{c_i}$ in the codebook,

$$\mathbf{c}_i = (c_{i1}, \ldots, c_{iN}) \tag{2.2}.$$

The matching of vector is based on distortion measure between codewords and input vector. The most common distortion measure of the quantizer is mean square error, which is defined by the Euclidean distance between vectors:

$$d\{\mathbf{s}, \mathbf{c}_i\} = \sum_{j=1}^{N} (s_j - c_{ij})^2 \tag{2.3}.$$

If the codewords $\mathbf{c_i}$ which is the most similar to vector $\mathbf{s}$ (and thus minimizes the distance $d\{\mathbf{s}, \mathbf{c_i}\}$) is found, the index of that codeword is coded and transmitted to the decoder. Figure 2.4 illustrates this idea. The decoding phase is a table look up procedure which used the received index to reconstruct the vector $\mathbf{c_i}$ at same codebook.

In order to obtain the optimal codebook, the true distribution of the source is necessary. If it is not known, which is generally the case, the LBG algorithms [55] can apply to find a suboptimal codebook based on a set of training samples. If a model of the image is known, then a synthetic generation of the codebook can be used [56].

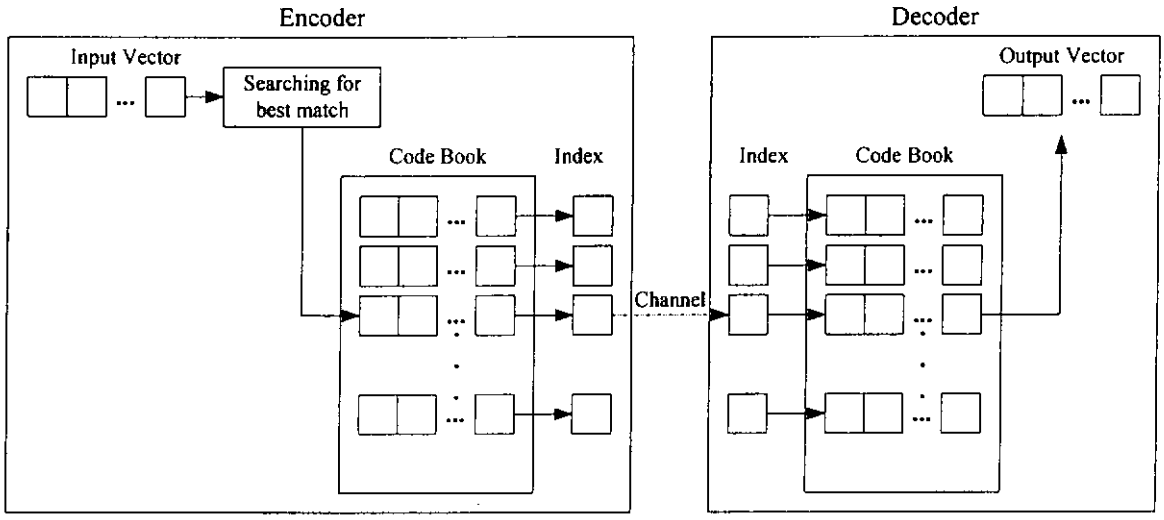


Fig. 2.4. A vector quantization encoding and decoding procedure.

2.2.6 Entropy coding

For most of the standard video coding schemes, entropy coding is applied to the output of quantizer. Indeed, entropy coding takes advantage of the statistical distribution of the output symbols. Some symbols can occur more frequently than the others and therefore we can use fewer bits for the coding of these symbols. One widely used entropy coding technique is Huffman coding [31].

Huffman codes belong into a family of codes with a variable length codeword. This means that individual symbols which make a message are encoded with bit sequences that have distinct length. It can be done because distinct symbols have distinct probabilities of incidence. This fact helps to create such codewords, which really contribute to decreasing of redundancy i.e. to data compression. Symbols with higher probabilities of incidence are coded with shorter codewords, while symbols with lower

probabilities are coded with longer codewords. Although, longer codewords cannot be prevented to show up, their effect on overall compression efficiency is still low because the probability of their occurrence is very low.

Several expansion of Huffman code can also be found. For example, adaptive Huffman code [51] enables dynamically change of the codewords accordingly to the change of probabilities of the symbols. In this way, the produced code is more effective then the primary Huffman code. Another enhancement of Huffman codes [52], have the characteristic that the coding scheme is coding group of symbols rather than a single symbol and it can prevent error propagation along compressed data which are corrupted by transmission error.

An alternative encoding method known as arithmetic coding has been developed [30]. In arithmetic coding, a message can be represented by an interval of real numbers between 0 and 1. As the message becomes longer, the interval needed to represent it becomes smaller, and the number of bits needed to specify that interval grows. Successive symbols of the message reduce the size of the interval in accordance with the symbol probabilities generated by the model. The more likely symbols reduce the range by less than the unlikely symbols and hence add fewer bits to the message. Instead of using predefined codeword tables for each symbol, arithmetic coder uses cumulated tables. These tables contain the cumulated frequencies for all possible symbols of every syntax element. The frequencies are determined by statistical measurements and must be proportional to the corresponding event probability. One table for each syntax element represents its probability distribution. Based on this model, the coder generates a bit stream for the incoming sequence of symbols. Hence it can combine subsequent symbols

and assign one codeword to them. Therefore, arithmetic codes permit non-integer number of bits to be assigned to each symbol and the symbols can be coded almost at their entropy rate [30].

In H.263, fixed statistical models of data source for arithmetic coder are used, that is all predefined symbol frequencies remain constant during encoding. However in real applications, the knowledge of symbol frequencies is not known and most probably the statistics are changeable with time. Hence an adaptive arithmetic coding is needed and several methods [53] and [54] are proposed for improvement of video compression using adaptive arithmetic coding.

2.3 Content-based techniques

2.3.1 Introduction

The term content-based coding denotes a scheme to separately encoded objects inside a video sequence. With this coding scheme, new functionalities such as content-based interactivity, hybrid natural and synthetic data coding for video application can be provided. Recent development of the new video coding standard MPEG-4 [12] is to support and to provide these content-based functionalities.

To provide these content-based functionalities, MPEG-4 relies on a content-based representation of audio-visual objects. It treats a video sequence as a composition of several objects that are separately encoded and decoded. This requires a prior decomposition of video sequence into semantically meaningful objects or so-called video

object planes (VOP's) [12]. Each frame of a video sequence is then composed of VOP's corresponding to the objects in the scene.

However, VOP decomposition is not a trivial task [33]. We need a proper definition of VOP so that only meaningful objects are segmented out. This can be done manually or by using chroma key technology. However, manual segmentation is a time consuming task while chroma key is limited to studio scenes.

Automatic or semiautomatic segmentation is another approach for VOP generation. For example, segmentation techniques using morphological operator [42] and [43] are suggested for generating VOP in an automatic manner. In these approaches, pixel intensity is used as feature for segmentation. It is noticed that such low-level segmentation algorithms fail to obtain meaningful objects in practical situation since objects of interest are often not homogeneous with respect to low-level features such as color and intensity.

2.3.2 Motion segmentation

Motion, on the other hand, is often used as a cue for segmentation. Physical objects are often characterized by a coherent motion that is different from that of the static background. By collecting motion information from the video, we can separate the objects out from the background. A classical approach to motion segmentation is to estimate a dense motion field followed by a segmentation of the scene based only on this motion information. However, the accuracy is limited to the estimation of motion field which is noise sensitive. In practice, it is difficult to group pixels into objects based on the similarity of their flow vectors as shown in Figure 2.5.

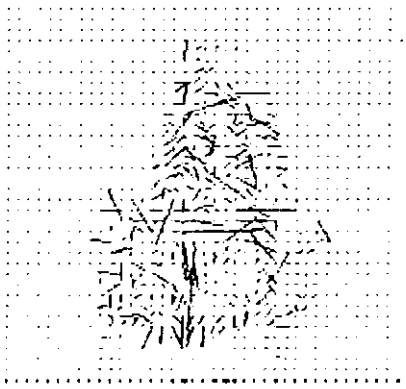


Fig. 2.5. Optical flow field estimated by the Horn-Schunck method [41] for frame 40 of the sequences Alexis. [33]

Rather than motion field, change detection mask (CDM) is another form of motion information that can be used in the segmentation process. In the mask, every pixel where the luminance of the image has changed due to a moving object is marked. In [34], an automatic and noise robust segmentation algorithm is suggested. Figure 2.6 shows this segmentation algorithm as a block diagram. First, an initial change detection mask between two successive frames is calculated, using a global threshold. Then the boundaries of changed image areas are smoothed by a local adaptive relaxation technique as shown in Equation 2.4. The local decision rule should read as: if squared luminance difference, d_1^2, exceeds the threshold term on the right hand side of Equation 2.4, the pixel is set to changed, otherwise it is set to unchanged.

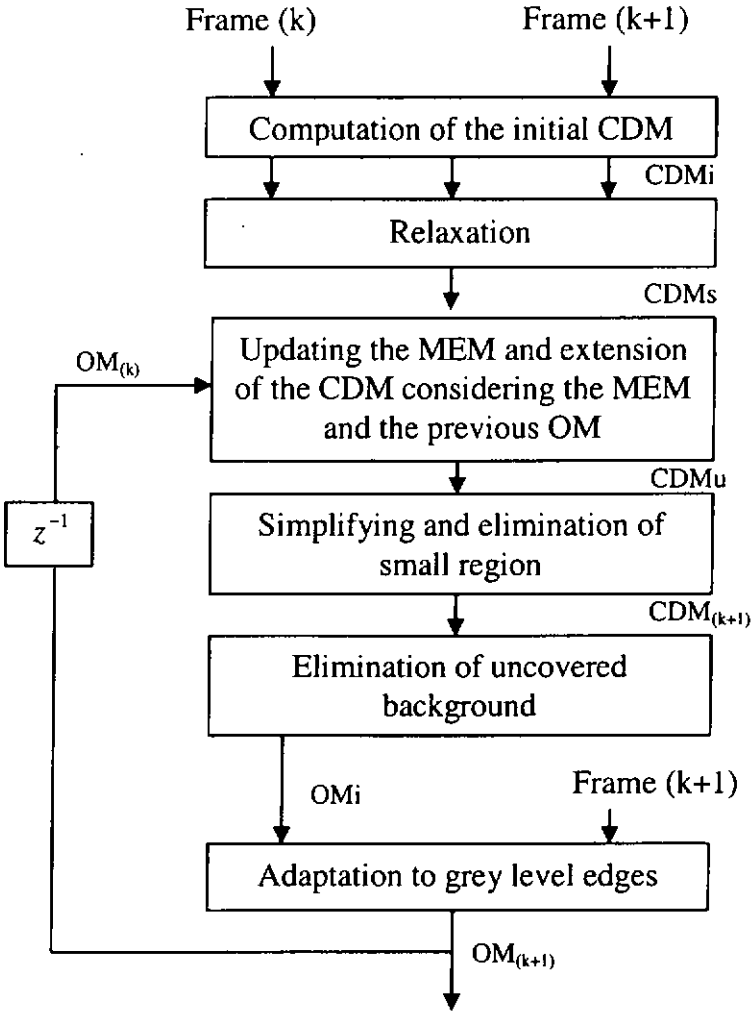


Fig. 2.6. Block diagram of the segmentation algorithm.

$$d_i^2 \underset{u}{\overset{c}{>}} 2\frac{\sigma_c^2\sigma^2}{\sigma_c^2-\sigma^2}\left(\ln\frac{\sigma_c}{\sigma}+(v_B(c)-v_B(u))B+(v_c(c)-v_c(u))C\right) \qquad (2.4)$$

where σ is equal to twice the variance of the assumed Gaussian camera noise distribution; σ_c is the variance of luminance differences within object regions; $v_B(q_k)$ denotes the number of horizontal and vertical neighbors of pixel k with the opposite label to q_k; and $v_c(q_k)$ denotes the number of diagonal neighbors of pixel k with the opposite label to q_k.

In order to obtain temporally stable object regions, the CDMs is updated with the previous object mask (OM). A pixel from the previous OM is labeled as changed if it was also labeled as changed in the CDMs of one of the last L frames. The current CDMs is then updated by a logical OR operation between CDMs and the previous OM and the resulting change detection mask CDMu is simplified by morphological closing and elimination of small regions.

The OM is then calculated from the CDM found in above steps by removing the uncovered background region. Finally, the boundaries of the resulting OM are further adapted to the grey level edges of the corresponding image in order to improve the accuracy.

As the decision of the change of pixels depends on the variances σ_c^2 and σ^2 which are updated on each frame, this segmentation algorithm is adaptive to the change of video content. The OM of the previous frame is used for calculating the variance σ_c^2 and the change detection mask of the previous frame is used for calculating the variance σ^2. In order to get more stable values for these variances, the currently measured values are averaged with the last three measured values. Thus, the algorithm gets more robust in case of noise.

Although the local adaptive relaxation technique can give a relatively accurate segmentation of objects, it is computation-intensive. When generating the CDMs, the threshold term on the right hand side of Equation 2.4 should be calculated, which requires the information of each pixel for the calculation of its variance. It can be a burden especially to real-time systems. Similar approach can also be found in [35], but the

segmentation mask is generated by comparing the differences of image frames with a threshold and followed by some morphological operations. It is simpler than [34] but with a trade-off in accuracy.

2.3.3 Semi-automatic segmentation

Motion can be used as a cue to separate physical objects from static background, but sometimes we still need to extract higher level information, such as speed, size and shape, from the separated objects to enhance the coding performance. However, to further classify the segmented objects is difficult for automatic segmentation approaches [33] unless a very constrained situation is present. Semi-automatic segmentation that requires user intervention in parameter setting can significantly improve the segmentation result [36] and [37]. This kind of semi-automatic segmentation techniques appears to be the most promising approach for general VOP segmentation [33].

2.3.4 Content-based coding

The segmented VOPs are then encoded separately which is so-called content-based coding. Content-based coding techniques allow accurate representation of objects such that we can efficiently allocate available bits for their coding. Besides, each specific object can be selectively encoded. Hence, at the decoder side, a subset of objects can be obtained from the bit stream. Due to the ease of bit allocation to each object, this technique is particularly useful for low bit rate video coding in which the channel bandwidth is narrow. Basically, the VOP coding algorithm involves at least two steps: 1) the coding and

transmission of the shape of the VOPs and 2) the coding and transmission of the pixel values insides the VOP. The shape information can be represented by chain coding [57], bitmap coding [12] or quadtree coding [58]. The pixel values or sometimes called textural content of the objects can be coded efficiency using transform-based techniques similar to those used in block-based methods [39], [35] and [40] or just using VQ techniques [38] .

For the content-based approach in [39], shape coding is first performed on the segmented input image. To represent the shape of the object, either chain code or plural rectangles is used. Figure 2.7a shows an example of chain coding and its chain code index is shown in Figure 2.7b. For a more simple shape such as that shown in Figure 2.8, plural rectangles are used to express the selected objects approximately which allows a great saving in the required coded bits. To remove the temporal redundancy between frames, each object is divided into macroblocks with size 16x16 pixels so that block-based motion compensated estimation can be implemented. DCT, shape adaptive DCT (SADCT) and variable length coder (VLC) are used to encode the residual error. SADCT is used to transform any residual error that with arbitrary shape.

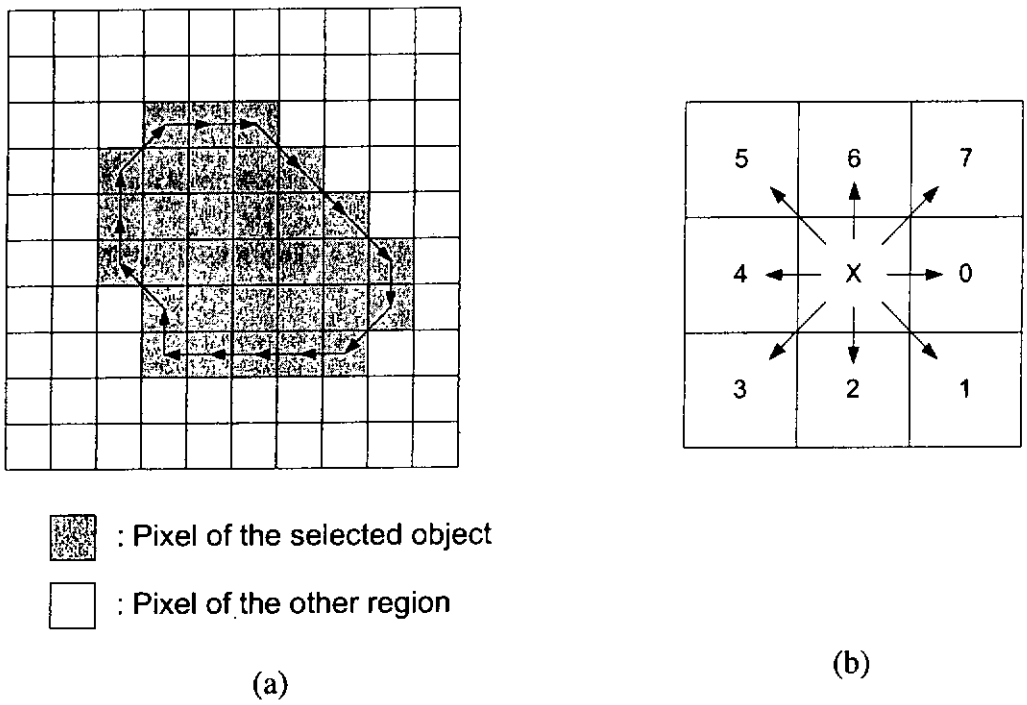


Fig. 2.7. (a) An example of chain codes and (b) the chain code.

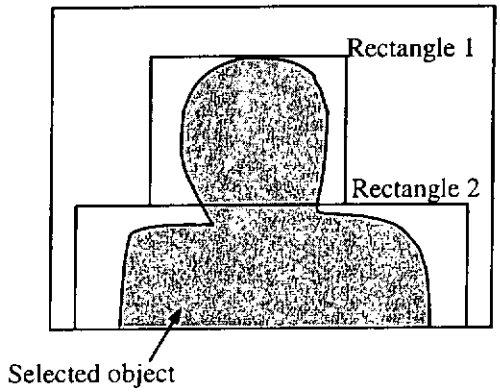


Fig. 2.8. Rectangular representation.

While in [35], the content-based video coder is shown in Figure 2.9. The segmentation unit is implemented by thresholding the absolute difference of current and previous images. The shape mask of object is formed which represents macroblocks that lie inside the object's region with a binary 1 and blocks outside the regions with a binary 0. The algorithm then exploits the temporal similarity of these block-based shape masks so

that the number of bits to represent a given shape in the current frame can be reduced. The author also suggested a spline-based shape representation to increase the accuracy of representation, however, it will increase the consumption of bandwidth. The segmented objects are then passed through the block-based motion compensated estimation unit to remove the temporal redundancy. To further reduce the amount of residual errors due to some sudden changes of motion, motion failure region detection is applied. The aim of this failure detection is to re-apply the segmentation procedure to the compensated image and the current image so that motion failure region can be extracted. The shape of the motion failure region is encoded using previous shape representation method while the content is encoded using DCT or wavelet transform.

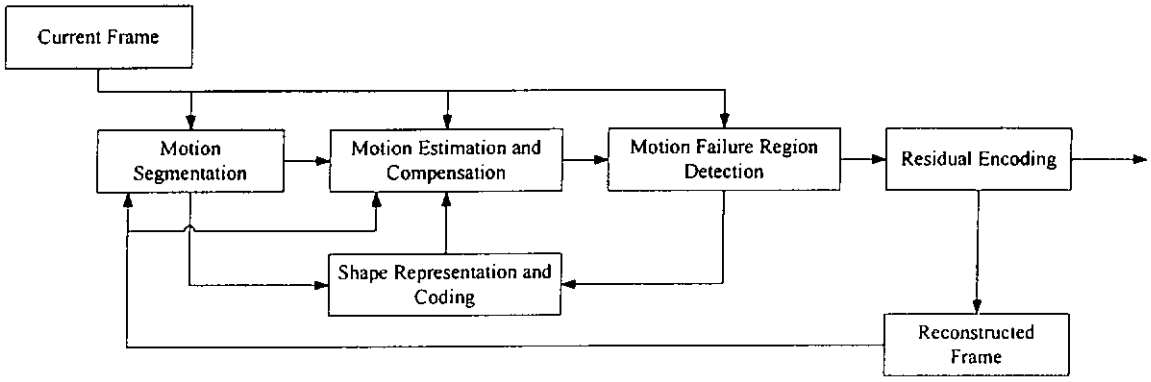


Fig. 2.9. The block diagram of the hybrid object-based coder [35].

Although the above content-based coding techniques have been proved to be efficient for low bit rate video coding, still, representation of object shape needs a large number of coded bits. It gives rise to a more simplified approach for shape representation which considers objects as a combination of image blocks. While most of the traditional video coding systems are block-based, the block-based representation techniques can naturally adapt to the current video coding systems. These techniques provide

computationally inexpensive and efficient techniques for very low bit rate video communications. Although block-based representation of objects cannot represent natural objects accurately, they are used in many content-based video coders [40].

For example, in [40], block-partitioning technique is used to get well balance of shape representation and overhead. From Figure 2.10, four patterns representing a vertical edge, a horizontal edge, a right-up diagonal edge and a left-up diagonal edge are used to roughly represent object shape. As the objects are represented in macroblock basis, it can be easily adopted to current video codec such as H.263. The video codec is shown in Figure 2.11. To improve the efficiency of motion prediction, two references frames are used. One is a reference stored in short-term frame memory (STFM) which is overwritten frame by frame. The other is a reference stored in long-term frame memory (LTFM). The LTFM works as background image memory. The current frame is first predicted from both LTFM and STFM inside block predictor and block-partition predictor, and the selector selects the final motion prediction mode. The selected motion prediction mode (MCMODE) and motion vectors are then encoded with VLC.

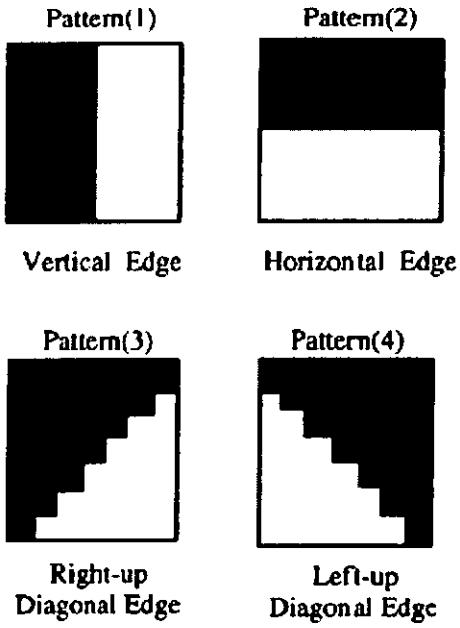


Fig. 2.10. Four types of block partitioning [40].

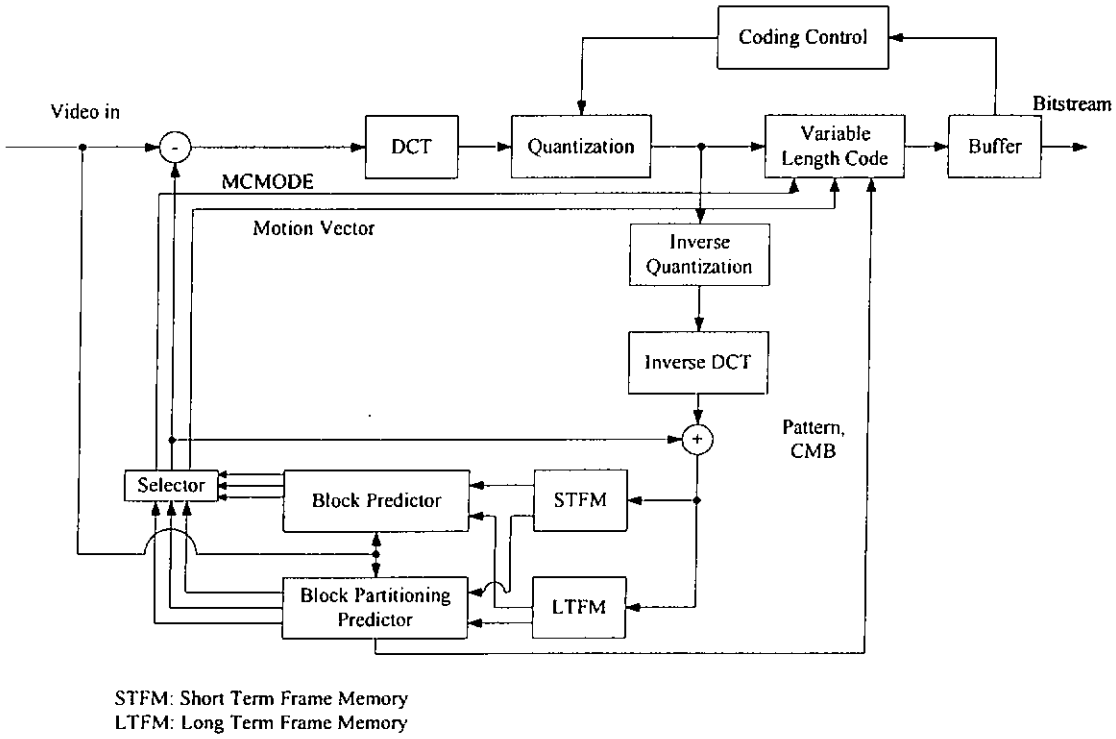


Fig. 2.11. Content-based video coder [40].

This scheme can save the bits required for object shape representation and improve in motion prediction. However, the codec requires additional codeword which is incompatible with current video coding standard. Besides, the additional motion estimation process for block-partition is computationally expensive.

2.4 Temporal scalability

2.4.1 Introduction

Apart from content-based coding, scalability is another feature of second generation video coding algorithms. A scalable video compression algorithm allows extraction of coded visual information at various data rates from a single compressed stream. It facilitates video services of different quality for different users under different constraints e.g. network bandwidth. There are a few approaches to achieve scalable video coding. One of the most commonly used approaches is the so-called temporally scalable video coding, which achieves scalability by embedding videos of different frame rates into a single bit stream. Consequently, users can extract video at the required frame rate according to their constraint.

The degree of temporal scalability and coding efficiency are a function of the number of frames in a group of frames (GOF), which is defined as the number of consecutive frames that can be decoded as a group independent from the rest of the video sequence. Temporal scalability is achieved by decoding subsets of the GOF consisting of equally spaced frames. Ideally, a temporally scalable coding technique should not only provide excellent visual quality at full frame rate but also maintain good visual quality at

lower frame rates. In fact, for lower frame rate video, the coding of motion is even more important as the video quality will easily be degraded if motion is not smooth enough at lower frame rate. The ability of maintaining good visual quality at various frame rates is very often determined by the interframe coding method chosen. Two general categories exist: predictive coding and subband coding. They are described in the following sections.

2.4.2 Motion compensated prediction

The motion compensated prediction (MCP) approach achieves temporal scalability by strategically placing reference frames when encoding the video. With the existence of reference frame, user can selectively decode a partial set of frames from the original set. The result is a temporal subsampling of the original sequence, and the decoded frames are exactly equal to those that would appear in the full frame rate video, although at a lower frame rate. There are two approaches for MCP coding. They are telescoping prediction and recursive prediction. In telescoping prediction, the first frame of GOF is used as a reference frame and the remaining frames are directly predicted from the first frame (as shown in Figure 2.12a). Once the reference frame is decoded, any of the remaining frames can be decoded also. This method provides a maximum flexibility on the temporal scalability, however, the coding efficiency decreases as the distance between the reference frame and predicted frames is large. A solution to this problem is using recursive prediction. As shown in Figure 2.12b, this method introduces several reference frames within a GOF. Each of the reference frames is predicted using either the first frame or another reference frame within the GOF. While the predicted frames are predicted from the nearest previous reference frame, hence higher coding efficiency can be achieved. However, the implementation of temporal scalability is not flexible enough when

comparing to the telescoping approach. As shown in Figure 2.12b, the degree of temporal scalability is discretized into full frame rate, 1/2 frame rate, 1/4 frame rate and 1/8 frame rate.

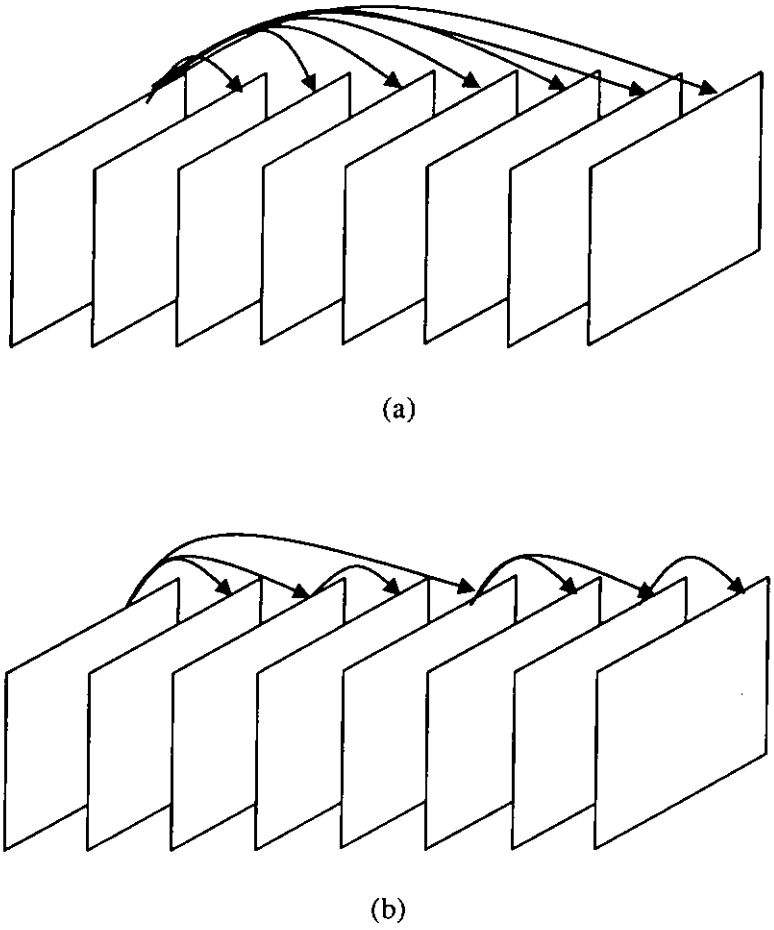


Fig. 2.12. Temporal prediction techniques that facilitate scalable video coding: (a) telescoping prediction and (b) recursive prediction.

2.4.3 TSB and motion compensated TSB

Temporal subband (TSB) coders and motion compensated TSB (MC-TSB) coders reduce the temporal redundancy of video by applying a subband or wavelet analysis in time [76], [77] and [32]. Figure 2.13 shows a typical framework for 3-D subband

decomposition. In the figure, two temporal subbands of the original video are created and several spatial subbands are also introduced in each temporal subband. TSB is invertible if the filters satisfy the perfect reconstruction property. Although any subband filters can be used, in nearly all practical coders two-band filters are used hierarchically. The most commonly used filters are the two-tap Haar wavelet filters, which are preferred because they consume less memory and introduce less coding delay and complexity than longer filters. In Figure 2.14 the terms *HP* and *LP* refer to high-pass filtering and low–pass filtering whereas the subscripts *t*, *h*, and *v* refer to temporal, horizontal and vertical filtering respectively. The output of Haar filters are basically the difference and average between frames, producing a high-pass and a low pass temporal frequency band respectively.

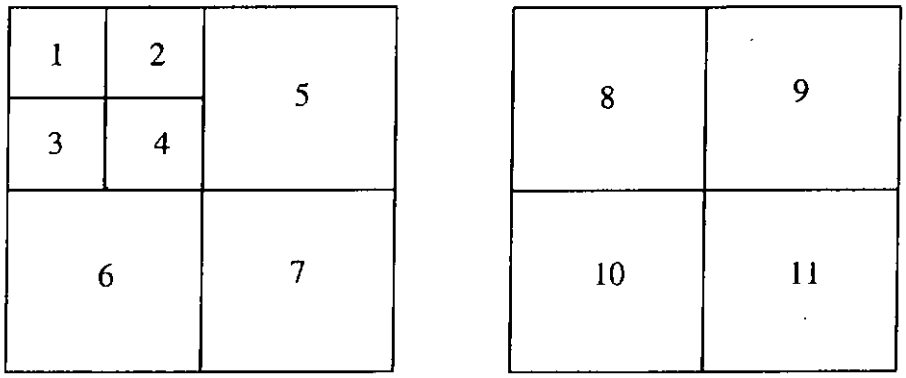


Fig. 2.13. Template for subband display.

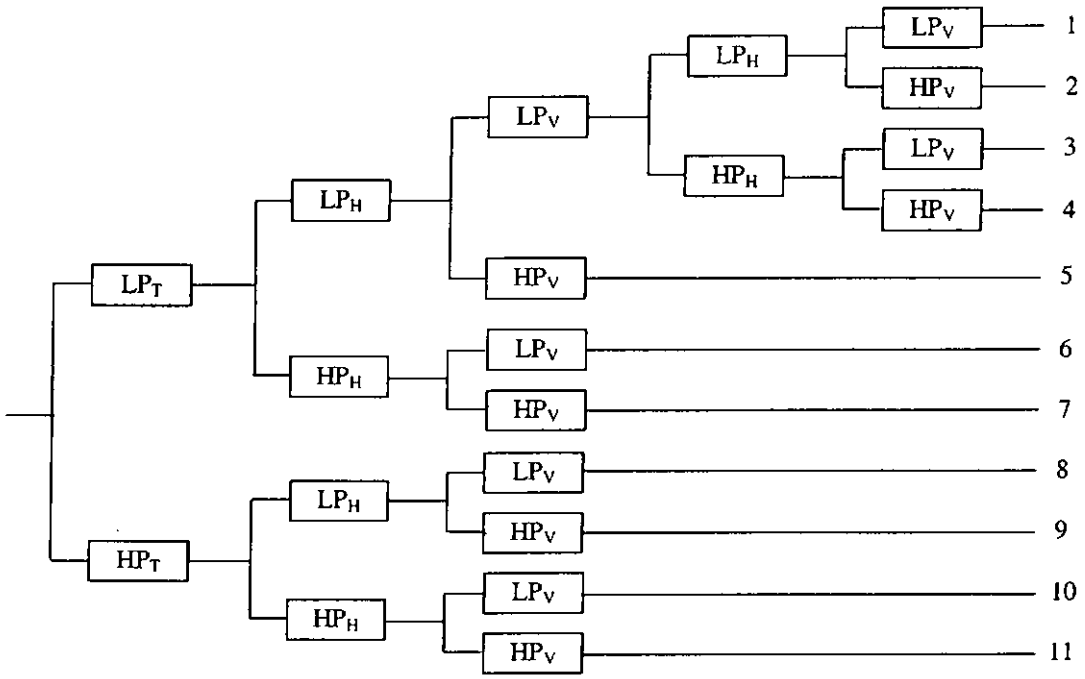


Fig. 2.14. Three-dimensional tree-structured decomposition.

While the high-pass temporal filters basically give the difference of consecutive video frames, it can be deduced that the information on the high-pass band can be minimized if the correlation between the consecutive frames is very high. In fact, we can further increase the correlation between frames by aligning equivalent scene content in time. It is achieved by applying motion compensation to temporal filtering process of the video frames. However, two issues are needed to consider for such implementation. Firstly, the order of applying motion compensation and temporal filtering process is considered. Coders can apply motion compensation once before applying temporal filtering or apply motion compensation prior to each stage of subband analysis. However, modeling motion along an entire GOF can be difficult, hence in [44], it is suggested that motion compensation is performed before the temporal filtering. The second issue we need to consider is the occlusion problem, which occurs when a one-to-one correspondence

between pixels in two frames does not exist in the motion compensation operation. There are two kinds of occlusions: covered and uncovered pixels. Covered regions are those regions that are found in the previous frame but not in the current frame; whereas uncovered regions are those regions that are found in the current frame but not in the previous frame. The predictive coding of covered and uncovered regions affects the efficiency of MC-TSB coding. If motion is modeled well by block-based motion compensation, the penalty on coding of such region is minimized. On the contrary, if motion cannot be modeled well on block level, many such regions would exist and the coding efficiency will drop.

One possible way to overcome this problem is predicting the overall motion of entire set of frames. This global motion compensation can provide a significant coding gain [45] and the occlusion problems only occur at edges of the frame. However, this method is limited to certain types of motion.

Instead of frame-based approach, block-based motion compensation has been combined with TSB coding [44]. In this scheme, motion compensation is performed on individual blocks prior to each application of the low-pass filter, and the local motion can be compensated well. The aligned video frames are then temporal filtered. However, the occlusion problems still happen and special provisions must be made. Hence in [49], the uncovered region are placed in the low-pass subband and coded directly. The covered regions are subtracted from a prediction and placed in the high pass subband.

In TSB and MC-TSB coding, the temporal scalability is achieved by decoding and synthesizing selected temporal subbands. This allows lower frame rates that halve with

each subband analysis level. However, in TSB, the low-pass subband filters basically perform as an averaging operator of the video frames. The result is a blurring of motion and the quality of video will decline as the frame rate decreases. Although the addition of motion compensation can improve the quality of lower frame rate video, the additional motion compensation process will increase the complexity for encoding process.

It is noticed that the wavelet basis used in the traditional TSB approach cannot give good quality video at lower frame rates. Hence in [50], an interpolating wavelet filter is used to achieve temporal scalability. It shows better performance in both full and lower frame rates video coding. By making use of the interpolation properties of the interpolating wavelet filters, the quality of the decoded video frame is the same for both full and lower frame rates. To understand that approach, first we need to know more about the characteristics of interpolating wavelet transform.

2.4.4 Interpolating wavelet transform

According to [46], an interpolating wavelet of order D with scaling functions ϕ that satisfy the following conditions:

1. Interpolation:

$$\phi(k) = \begin{cases} 1, & k = 0 \\ 0, & k \neq 0 \end{cases} \quad k \in Z \tag{2.5}$$

2. Self-Induced Two-Scale Relation: ϕ can be represented as a linear combination of the dilates and translates of itself, while the weight is the value of ϕ at a subdivision integer of order 2

$$\phi(x) = \sum_k \phi(k/2)\phi(2x-k). \quad (2.6).$$

3. Polynomial Span: For an integer $D \geq 0$, the collection of formal sums, symbolized by $\sum C_k \phi(x-k)$ contains all polynomials of degree D and C_k is filter coefficient.
4. Regularity: For real $V > 0$, ϕ is Hőlder continuous of order V.
5. Localization: ϕ and all its derivatives through order $\lfloor V \rfloor$ decay rapidly:

$$\left|\phi^{(r)}(x)\right| \leq A_S (1+|x|)^{-s}, \qquad x \in R, \qquad s > 0, \qquad 0 \leq r \leq \lfloor V \rfloor \quad (2.7)$$

here $\lfloor V \rfloor$ represents the maximum integer which does not exceed V and A_s is a constant.

The interpolating wavelets have many useful properties such as FIR filter bank realization, linear phase, and convenient boundary filter design. Comparing with the commonly used wavelet transforms, the interpolating wavelets possess the attractive characteristic that the wavelet coefficients are obtained from the direct linear combinations of discrete samples rather than from the traditional inner product integrals [46]. Besides, the coefficients can be calculated in a parallel computation manner. For the halfband filter with length L, the calculation of the wavelet coefficients does not involve more than $L+2$ multiply-adds for each coefficient. For a D-times differentiable function, the localization property of the interpolating wavelet coefficients is comparable to the property of coefficients of smooth orthogonal wavelet decompositions.

There are 2 well-known families of such wavelets: the interpolating spline wavelets and the Deslauriers-Dubuc interpolating wavelets [47]. We consider here the second family of interpolating wavelets since the interpolating spline wavelets do not have

compact support. To obtain compactly supported interpolating wavelets, the Deslauriers-Dubuc's "fundamental functions" are frequently chosen as the scaling functions, which satisfy the two-scale equation as follows:

$$\varphi(x) = \varphi(2x) + \sum_k h(k)\varphi(2x - 2k + 1) \tag{2.8}$$

where h is an interpolating filter, which can be obtained using the Lagrange interpolation formula. Based on the Deslauriers-Dubuc interpolating scaling function, Donoho constructed a family of interpolating wavelets [46]:

$$\varphi(x) = \varphi(2x) + \sum_k h(k)\varphi(2x - 2k + 1) \tag{2.9}$$

$$\psi(x) = 2\varphi(2x - 1) \tag{2.10}$$

with the duals

$$\tilde{\varphi}(x) = \delta(x) \tag{2.11}$$

$$\tilde{\psi}(x) = \tilde{\varphi}(2x - 1) - \sum_k h(-k)\varphi(2x - 2k - 2) \tag{2.12}$$

where $\delta(x)$ denotes the Dirac impulse, $\tilde{\varphi}$ and $\tilde{\psi}$ are the dual scaling and wavelet functions, respectively.

However, Donoho's interpolating wavelets are non-orthogonal and have less degree of freedom for filter optimization. Later on, Sweldens proposed another interpolating wavelet which is a generalization of the biorthogonal interpolating wavelet using lifting scheme [48]. In this way the system can be represented as follows:

$$\varphi(x) = \varphi(2x) + \sum_k h(k)\varphi(2x - 2k + 1) \tag{2.13}$$

$$\psi(x) = 2\varphi(2x - 1) - \sum_k g(k)\varphi(x - k) \tag{2.14}$$

with the duals

$$\tilde{\varphi}(x) = 2\tilde{\varphi}(2x) + \sum_k g(-k)\tilde{\psi}(x-k) \tag{2.15}$$

$$\tilde{\psi}(x) = \tilde{\varphi}(2x-1) - \sum_k h(-k)\tilde{\varphi}(2x-2k-2) \tag{2.16}$$

where h and g are two interpolating filters and have only finite non-zero coefficients. The scaling function and wavelet function for such system can now be fully determined by the interpolating filters h and g and are flexible in design.

2.5 Summary

In this chapter, we have reviewed some of the basic video compression techniques that are used in the current video coding standards. As for the second generation video coding systems, two important functionalities, content-based coding and temporal scalability are reviewed. A detailed description on the theory and implementation of these two functionalities is also provided. It can be seen that problems still exist for the implementation of these two functionalities. In the following chapters, we propose several methods to resolve the problems.

Chapter 3

Real-time video segmentation for content-based coding

3.1 Introduction

The concept of content-based coding is suggested in recent video coding standards, such as MPEG-4. It is useful for low bit rate video coding when applicable bandwidth is not large enough to transmit the image sequence with good quality. With content-based coding, bit allocation is allowed to coarse code the details of some regions, e.g. static background, in order to save the budget for the coding of other regions with more vivid motion. However, to provide this functionality, we need a content-based representation of visual objects that treats a scene as a composition of several semantically meaningful object planes, which are known as video object planes (VOP's) [12].

In many cases, decomposing video into several meaningful VOP's is very difficult. And in most cases, some sort of preprocessing procedures must be performed to clearly define a VOP. They are known as video object segmentation [12], which is often the first step of a content-based video coding algorithm. Once a VOP is obtained, the next step is to extract the feature of the VOP such that bit allocation can be performed. A typical object feature that is required for most of the content-based video coding algorithms is the

motion of the object. It is because, in general, the faster the objects are moving, the larger the bit budget is required for their coding. After extracting the object features, we need to apply these features to the bit allocation process of the video codec. It is a difficult task since most of the current standard video codecs, including MPEG-1/2, H.263, etc. do not directly support content-based coding. Even if we obtain the object features, we may need to re-design the video codec in order to adopt this information for bit allocation. Very often, either the bit stream format needs to be changed or a drastic modification to the control unit of the video codec may be required.

In this chapter, we propose a practical content-based scalable video coding algorithm and apply it to the H.263 video coding standard. For coding video data for very low bit rate networks, H.263 video codec is recommended as one of the possible candidates due to its relatively simple structure and high coding efficiency [5]. However, it is expected that its performance can be further improved if the content-based scalable video coding technique can be applied. The major contributions of the proposed algorithm are (i) the introduction of a real-time video segmentation unit that automatically identifies the regions that contain moving objects or static background and (ii) the modification of the control unit of H.263 encoder to allow bit allocation according to object motion. The schematic diagram of the modified H.263 encoder is shown in Figure 3.1 where the modified functional blocks are shadowed. Although the H.263 encoder is modified to incorporate the content-based functionality, the encoded sequence is still completely comprehensible to the conventional H.263 decoder. As compared with the conventional H.263 codec, the content-based H.263 coding algorithm gives a consistently improvement for decoded video, in terms of peak signal to noise ratio (PSNR), at the same bit rate.

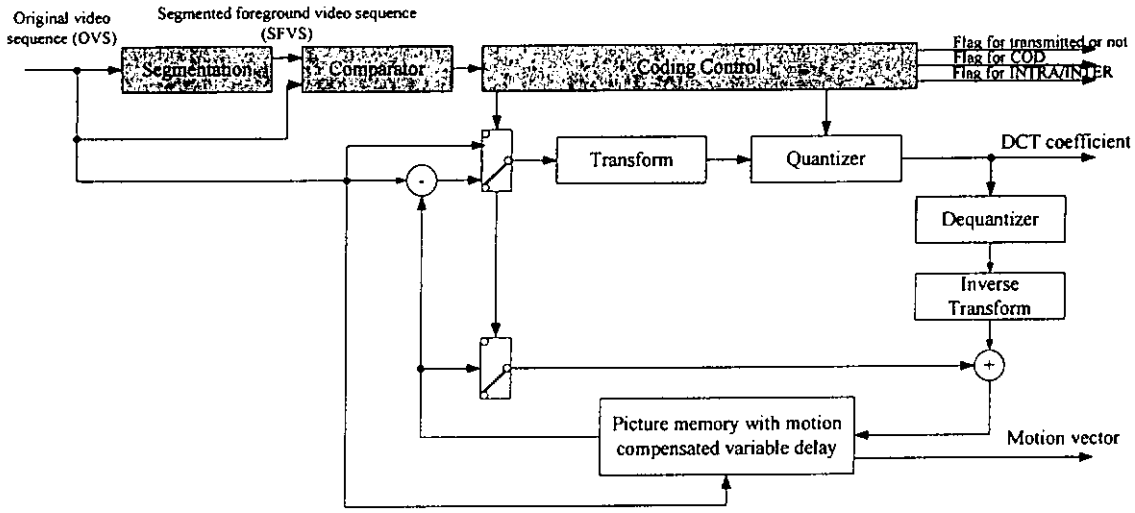


Fig. 3.1. The schematic diagram of the improved H.263 encoder.

3.2 *Video object segmentation*

There are several methods, in literature, suggested for object segmentation. For single frame, object segmentation schemes can be classified into two categories, boundary-based segmentation and region-based segmentation. By means of edge detection, boundary segmentation schemes try to locate the position of object edge [62]. The accuracy for the segmentation is dependent on the design of edge filter. As for region-based segmentation, neighborhood pixels with similar color or texture information [63] will be collected to form a region. However, both boundary-based and region-based may divide same object into different regions or group different objects to the same region if the objects have similar characteristic.

For video, motion often acts as a cue for segmentation. A classical approach to motion segmentation is based on the dense motion field estimation. Motion vector from

motion estimation is used to segment object with coherent motion relative to background [64]. However, the accuracy is limited to the estimation of motion field which is noise sensitivity and it is difficult to group pixels into objects based on the similarity of their flow vectors as shown in Figure 2.3 in Chapter 2.

For most of the low bit rate video applications, such as video conferencing or surveillance, without panning and zooming, the scene often has a fixed background and meaningful objects are often changing their spatial positions across time. Hence object segmentation can be achieved by simply comparing the consecutive video frames and detecting the changing regions in the scene. This is the basic idea of the Change Detection Mask (CDM) technique [34]. Based on the design of the comparison unit, the change detection schemes are classified into pixel-based, block-based and region-based [65]. While the result of pixel-based approach is susceptible to noise problem, the region-based approach can hardly be applied directly to traditional low bit rate video coding applications since additional bit budget for shape coding is required. Block-based, on the other hand, can outperform in dealing with noise and can be applied to current block-based video coding schemes without any drastic modification of the codec.

Based on the idea of using the CDM, there were many approaches proposed recently for video object segmentation. In [59], segmentation of moving objects in video sequence is achieved by thresholding the luminance difference between two successive frames. A more complicated segmentation method was proposed in [34], where a local relaxation technique is used to estimate the shape of the moving objects. By considering the estimated displacement vector field, areas of uncovered background on the CDM are removed and the boundary of object mask is improved by applying a grey level edge

adaptation technique and an object mask memory. However, the local adaptive relaxation technique is computation-intensive. For every border pixel k in the CDM, the local threshold is recalculated and compared with the squared luminance difference at the location of pixel k before making the decision on whether pixel k belongs to the changed or unchanged area. This time consuming procedure may not be suitable to real-time systems where a much simpler and effective segmentation algorithm will be more desirable.

3.2.1 Proposed segmentation algorithm

In this section, we propose a real-time segmentation algorithm that follows [59] but with some modifications. Figure 3.2 shows the flow diagram of the segmentation algorithm. Based on the luminance difference between two successive frames, an initial change detection mask (CDMi) is obtained by applying a global threshold, Th_{ch}, as shown in the first block of Figure 3.2. The locations with luminance difference greater than the threshold are set to 1, or otherwise 0, in the CDMi. The boundaries of the CDMi are smoothed by the morphological operators, opening and closing, and results in another mask, namely, CDMs.

As shown in Figure 3.2, the CDMs obtained after opening and closing are mapped into 8x8 data blocks. It is because the segmented video sequence will finally be sent to the H.263 video encoder, which is a block-based encoder. The block-based segmentation algorithm used here avoids the complicated shape coding procedure of the object boundary in each macroblock. Within each block, the number of CDMs is counted. If that value is larger than a threshold, Th_{mb}, that block will be assumed to be an initial motion

block (MBi). The choice of the size of MBi requires further elaboration. Since the size of each macroblock in H.263 is 16x16 pixels, it is straightforward to choose the size of MBi to be also 16x16 pixels. Nevertheless, it is noticed that using 16x16 pixels MBi will introduce too much background into each motion block. The next appropriate choice is to use 8x8 pixels MBi since each macroblock contains four 8x8 pixels blocks in H.263. By using 8x8 pixels MBi, it is noticed that the moving objects can be extracted more accurately. The result of using 16x16 and 8x8 pixels MBi is shown in Figures 3.3 and 3.4.

It is possible that the MBi obtained may not contain the whole part of the moving object. To avoid the missing of the corner part of the moving objects, we consider also the neighboring non-motion blocks of MBi since they have a high probability to be the motion blocks. This is the so-called region growing. By setting those non-motion blocks back to motion blocks, nearly all parts of the moving objects will be obtained and are located within the extracted motion blocks.

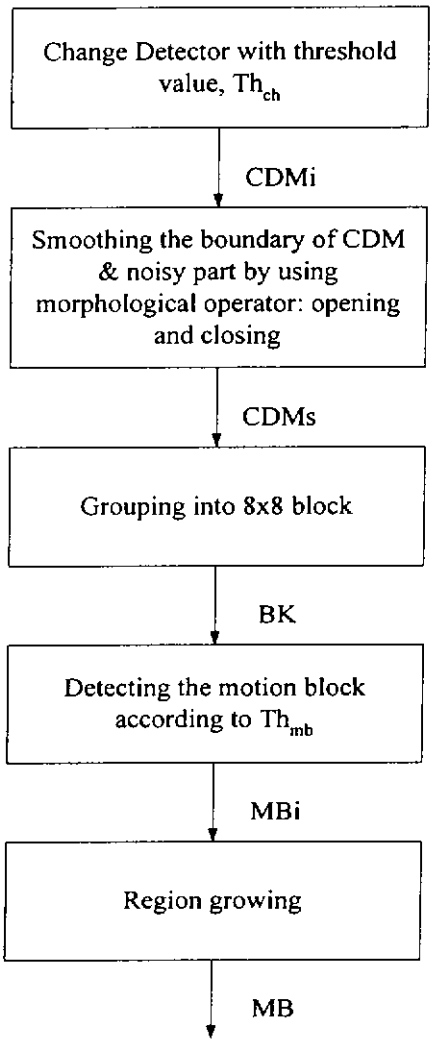
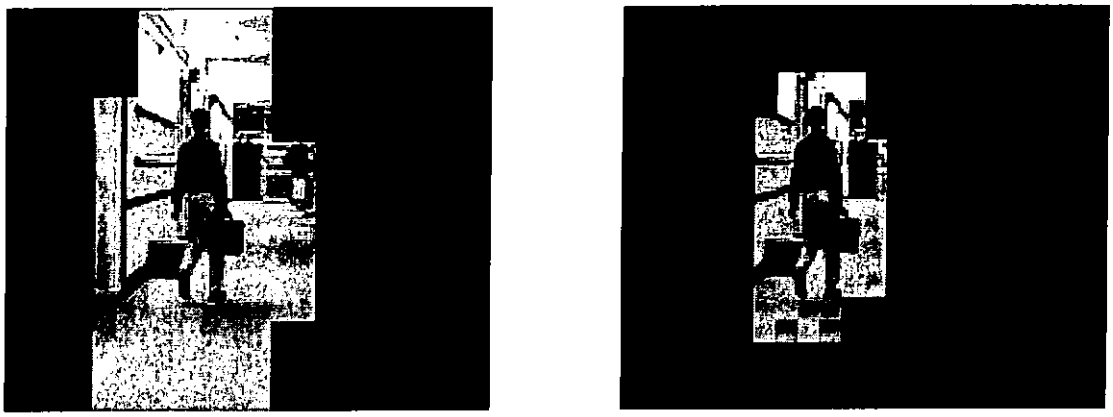


Fig. 3.2. Block diagram of motion block segmentation algorithm.



(a) (b)

Fig. 3.3. Extracted motion blocks of *Hall* in 16x16 pixels (a). Extracted motion blocks in 8x8 pixels (b).

(a) (b)

Fig. 3.4. Extracted motion blocks of *Salesman* in 16x16 pixels (a). Extracted motion blocks in 8x8 pixels (b).

3.3 On combining segmentation algorithm with H.263 video coding

Many methods have been suggested in literature for the coding of segmented video sequences. In [40], a special motion compensated technique was proposed. This technique uses block partitioning prediction rather than block based prediction. In this approach, each macroblock is separated into two partitions, motion compensation is then performed on each partition and motion vectors are encoded individually. However, this method

requires the introduction of new codewords to the standard of H.263 and the video bit stream generated is not comprehensible to the current H.263 decoder. Apart from adding new codewords in the encoder, another method [60] is suggested to develop a proprietary encoder for low bit rate coding. However, this introduces even more problems in applying the encoder to the current video transmission systems. In this section, we propose a modified H.263 encoder that can handle the segmented video sequence. No modification is required to the bit stream format hence the encoded bit stream is completely comprehensible by the current H.263 decoder.

To achieve content-based coding, we need the encoder to be able to allocate different amount of bits for the coding of different regions accordingly to their content. However, H.263 does not directly support content-based coding. We cannot explicitly instruct a H.263 encoder to allocate different bit budget to different regions. As an alternative, we adjust the coding rate of each macroblock to achieve bit allocation. More specifically, for the proposed approach, we first obtain by using the real-time object segmentation algorithm as mentioned in Section 3.2 the regions that contain foreground moving objects and static background. For the static regions of each video frame, their encoding process is skipped hence the bit budget for their coding is reduced.

As shown in Figure 3.1, the control unit of the proposed encoder makes use of the comparison result of 2 video inputs: (i) the original video sequence (OVS) and (ii) the segmented foreground video sequence (SFVS) obtained from the segmentation unit. The encoding procedure is similar to that for the traditional H.263 encoder, however, the control unit should refer to the comparison result when encoding the OVS as shown in Figure 3.5. If a MB of SFVS and OVS at the same location of the same frame are different

to each other, it is possible that some or all the blocks inside this MB are the static background blocks. The coding control should decide whether this MB should be encoded or not.

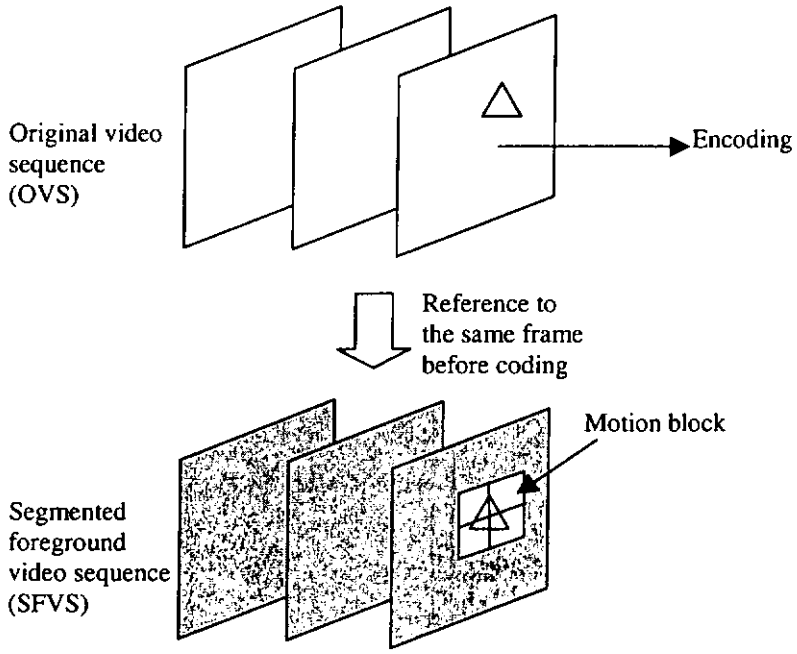


Fig. 3.5. The SFVS is used as a reference for encoding the OVS.

Although the above procedure seems simple, there are a number of problems need to be resolved during actual implementation. As mentioned before, the size of each motion block is set as 8x8 pixels. However, there is no mechanism in H.263 to skip the coding of an individual 8x8 pixels block within a MB. Instead, we can only skip the coding of the whole MB by setting the "coded macroblock indication" (COD) to '1'. To solve this problem, we make use of the Advanced Prediction Mode (Annex F) of H.263. In this mode, four motion vectors for the four blocks, respectively, of a MB are generated. In this case, if a particular block of a MB is the static background, the motion vector value and the "coded block pattern for luminance" (CBPY) for that block can be set to zero to reduce

the transmission data required. That is, the non-moving block is coded with zero motion vector and without any non-INTRADC coefficient.

Consequently, the following coding decision rules are implemented in the control unit of the encoder:

Case 1: If the whole MB is static background as indicated in SFVS, the encoding process of this MB should be skipped by setting COD equals to '1'.

Case 2: If not all blocks in the MB is static background as indicated in SFVS, the control unit should identify which are the static background blocks. For the normal object blocks, the encoder should encode them in the normal manner. For the static background blocks, the encoder should encode the blocks with zero motion vector and set the CBPY of that block to zero.

Case 3: If the whole MB is an object as indicated in SFVS, the encoder encodes the MB in the normal manner.

The above decision rules will only be applied to the coding of INTER frames. For INTRA frames, they will be encoded as normal to allow a periodic refresh of the content in the decoded video. It is important particularly for video communications over error prone channels.

3.4 Encoding results on some standard sequences

We implemented the proposed system with a 450MHz Pentium II personal computer and compared with a conventional H.263 encoder, tmn3.2 [61]. Two standard QCIF video sequences, *Hall* and *Salesman* were used in the comparison. The sequences

are encoded at 0.03 bit per pixel (bpp) and 20 frames per second (fps), the peak signal to noise ratios (PSNR) of the decoded sequences as compared with the original ones are shown in Figures 3.7 and 3.8. It can be seen that, with the same bit rate, the PSNR of the proposed method is consistently higher than the original H.263 codec.

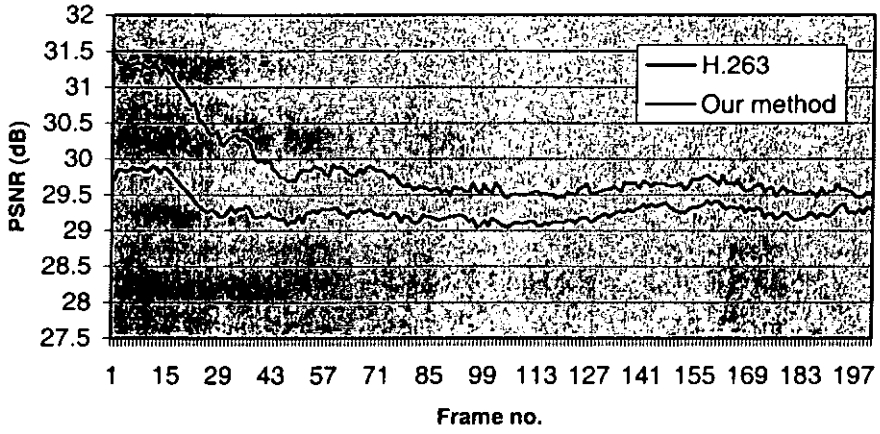


Fig. 3.6. A plot of PSNR against the frame no. for video sequences *Hall*.

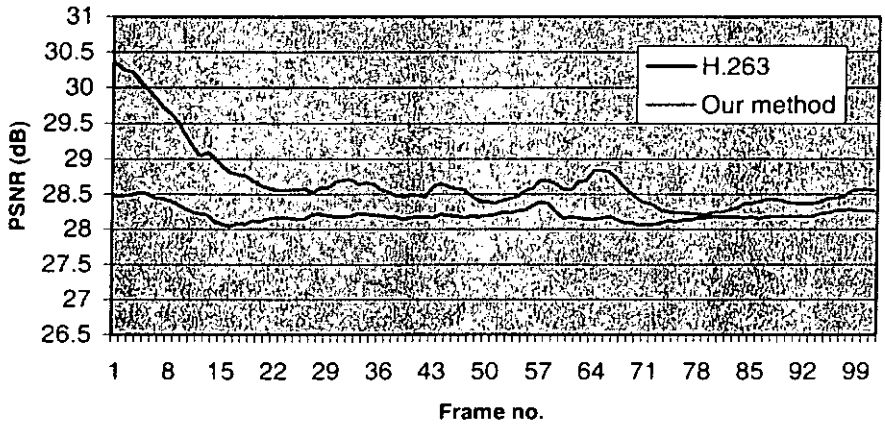


Fig. 3.7. A plot of PSNR against the frame no. for video sequences *Salesman*.

(a) (b)

Fig. 3.8. (a) The decoded frame (frame 25) of the original H.263 bit stream (*Hall*). (b) The same decoded frame that is encoded by the proposed system.

(a) (b)

Fig. 3.9. (a) The decoded frame (frame 10) of the original H.263 bit stream (*Salesman*). (b) The same decoded frame that is encoded by the proposed system.

Figures 3.8 and 3.9 show one of the decoded frames for the two standard video sequences using different systems, respectively. It can be seen that the quality of frames encoded by the proposed system is subjectively better, especially for moving objects. It is expectable since, as the total amount of bits allocated for each sequence is the same for both systems, the reduction in bit budget for coding the static background implies the increase in bit budget to refine the quality in coding the moving objects.

3.5 Summary

In this chapter, we have presented a content-based scalable H.263 video coding system that is suitable for low bit rate video applications. It has been shown that under a low bit rate condition, the proposed system can provide a consistent improvement in terms of PSNR when comparing with the conventional H.263 codec. Besides, the encoded video sequence can be decoded with conventional H.263 decoder which is different from other proprietary codec that requires a tailor-made decoder. The proposed system can be applied to low bit rate video applications, such as video conferencing systems and video surveillance systems with fixed camera setting.

While the proposed system allocates different amount of bits to the coding of foreground moving objects and static background, it is more desirable if we can further identify the moving objects as fast moving objects and slow moving objects, and allocate different bit budgets to their coding. To achieve this, we need a more sophisticated feature extraction technique to classify the speed of the objects in real-time. This kind of technique is often application oriented. In the next chapter, we demonstrate how it is achieved for road traffic surveillance systems.

Chapter 4

Application of content-based scalable H.263 video coding in road traffic monitoring

4.1 Introduction

In this chapter, we further extend the real-time video segmentation method, as discussed in Chapter 3, to real-time content-based video coding applications. As it is mentioned in Chapter 3, object segmentation and feature extraction are the kernel of content-based video coding. While it is relatively easy to distinguish moving objects from static background, higher level of object features are often required to optimize the coding performance. For example, it is desirable if the object speed is known, since in general the slower the object is moving, the lower the frame rate is required for its coding. To classify object speed from a video is never a trivial task. First, the term "object speed" needs to be carefully defined. For example, if the arms of a man are moving very fast, we cannot classify the man is moving very fast since perhaps his body does not move. While it is impossible to give general definition of higher level object features, their extraction and classification become application oriented. In this chapter, we particularly consider the application of content-based video coding to road traffic surveillance systems.

Recent advance in third generation mobile communication systems enables the transmission of video information using mobile channels. One of the possible applications of such systems is real-time road traffic monitoring using the mobile videophone. By having such system, road users can have a thorough understanding of road traffic condition and make judicious choices when selecting the paths to their destinations. We particularly interest in such application because first, such kind of service can effectively relief the road traffic problem in modern cities; second, it is a good platform to demonstrate the merits of content-based video coding.

As compared with other conventional applications of mobile videophone, such as, video conferencing, real-time road traffic monitoring inherently requires a larger channel bandwidth since the motion of cars on the roads is often much faster than the moving objects in a video conference. Nevertheless, video taken by the road traffic surveillance systems often has a fixed background without panning and zooming. This allows a simple segmentation of the moving objects (i.e. cars) and the static background (i.e. road), from the original video. Furthermore, since object (i.e. car) in a road traffic video often moves as an independent entity and on a restricted area (i.e. roads) of each of the video frame, it is easier to classify its speed and allocate suitable amount of resource for its coding.

In this chapter, we propose a content-based scalable video codec that particularly suitable for road traffic surveillance systems. The proposed codec is a modification to the ITU H.263 video codec, however, the encoded bit stream is comprehensible to the traditional H.263 decoder. For the proposed approach, input road traffic video is first processed by a real-time segmentation unit that the image blocks where moving objects are found are separated from those with steady background. Those image blocks with

moving objects are further analyzed and classified as high or low activity by assessing the regularity of their activities using the proposed correlation approach or zero crossing density detection approach. An image block is then coded in one of the 3 different frame rates according to its activity. The frame rate control is achieved externally by adaptively adjusting a threshold value of the segmentation unit. This avoids a drastic modification of the internal control mechanism of the H.263 encoder for implementing the complicated control logic. Besides the reduction of temporal redundancy, a psychovisual thresholding unit is introduced into the encoder to further reduce the spatial redundancy of image blocks. It is possible since human beings are more sensitive to slow moving objects. The schematic diagram of the improved H.263 encoder is shown in Figure 4.1 where the modified functional blocks are shadowed. The advantage of the proposed system is that the encoding method is based on the ITU H.263 standard and the encoded bit stream is completely comprehensible to the conventional H.263 decoder. As compared with the conventional H.263 codec, the proposed system improves the compression rate by more than 20% with negligible visual distortion.

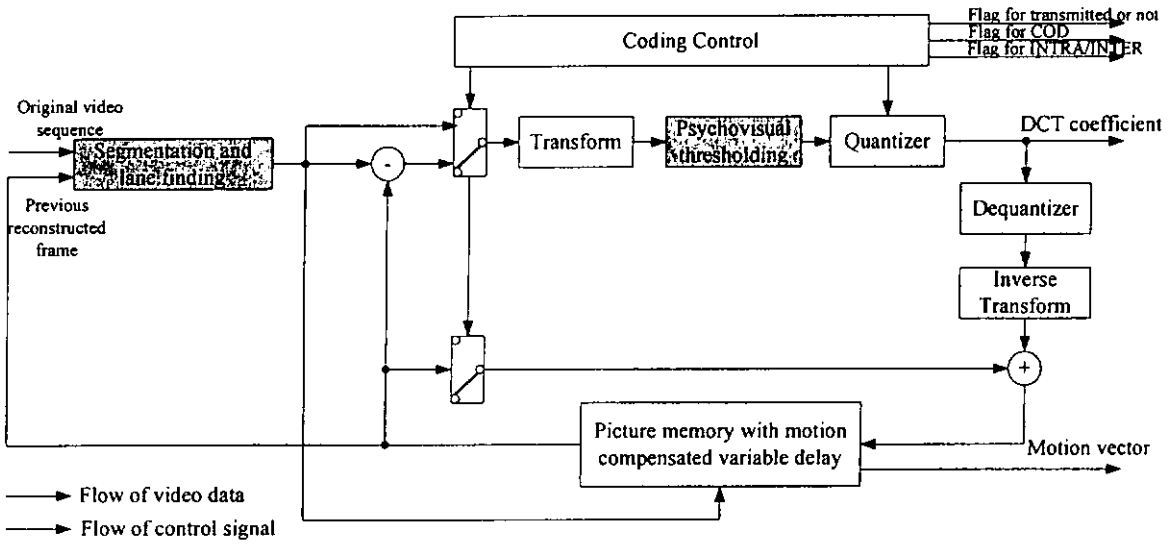


Fig. 4.1. The schematic diagram of the improved H.263 encoder.

4.2 The segmentation unit

To achieve content-based video coding, object segmentation is the first step that has to be performed. It is interesting to note that most of the road traffic video monitoring systems do not have fast panning and zooming functions. This simplifies the design of the segmentation unit for extracting the moving objects in the video. We have suggested in previous chapter an improved segmentation unit which is a block-based algorithm and can be applied to H.263. In this chapter, we apply this segmentation algorithm to our proposed video coding scheme. As mentioned in the previous chapter, using 16x16 pixels MB_i will introduce too much background into each motion block. The next appropriate choice is to use 8x8 pixels MB_i since each macroblock contains four 8x8 pixels blocks in H.263. By using 8x8 pixels MB_i, it is noticed that the moving objects can be extracted out more accurately. The result of using 16x16 and 8x8 pixels MB_i is shown in Figure 4.2.

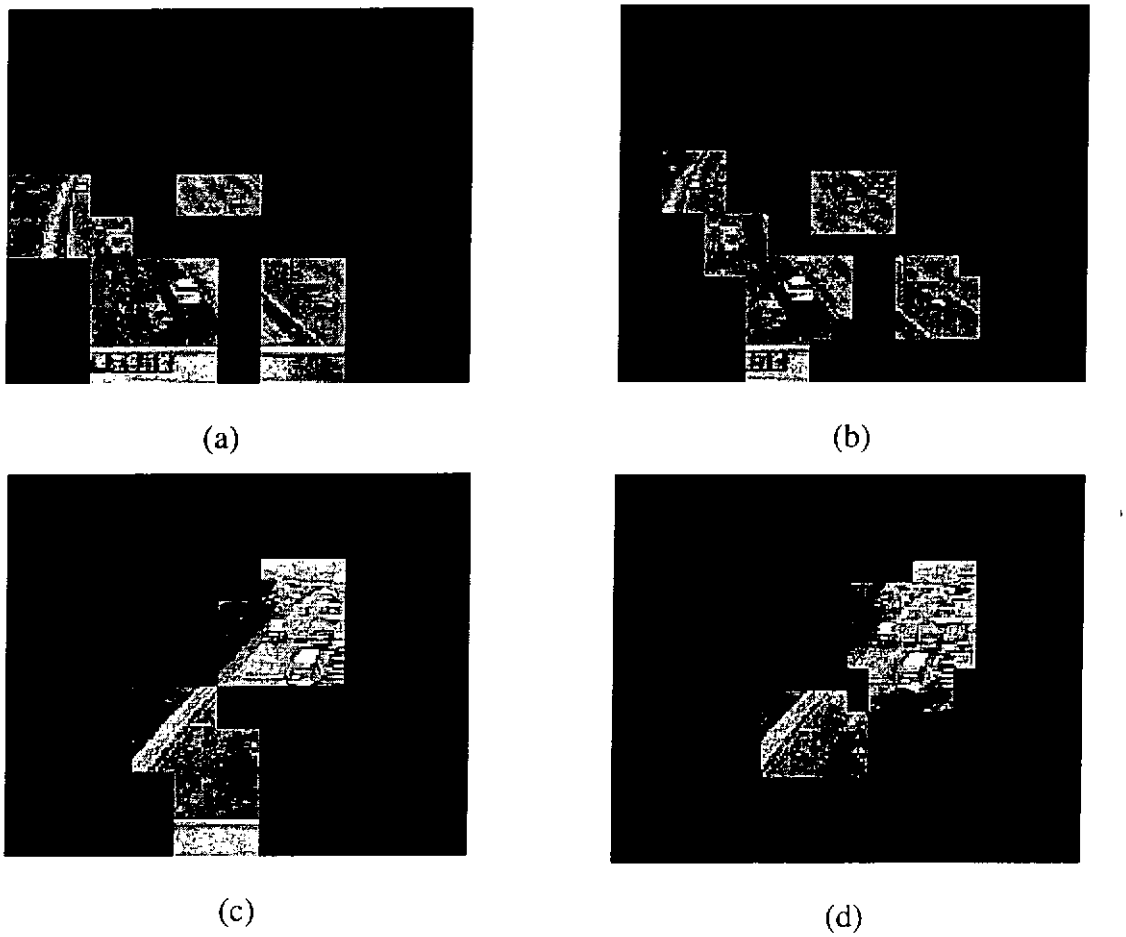


Fig. 4.2. Extracted motion blocks in 16x16 pixels (a) and (c); Extracted motion blocks in 8x8 pixels (b) and (d).

4.3 Motion detection according to lane finding

Although the above segmentation algorithm can successively extract the motion blocks from video sequence, the speed of the moving objects cannot be measured. This parameter is important since it can help us to better allocate the resource for coding. For example, it is obvious that the coding frame rate of a fast moving object should be higher than a slow moving object; while the resolution in coding a fast moving object can be lower than a slow moving object due to human perception. By carefully allocating

resource to the coding of moving objects of different speed, the spatial and temporal redundancies can be further reduced to achieve a higher compression efficiency.

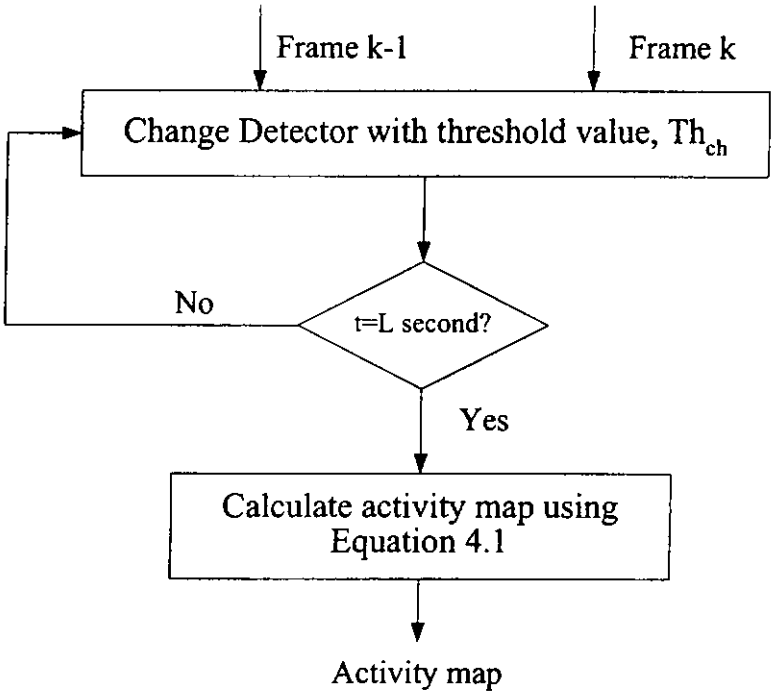


Fig. 4.3. Lane finding algorithm.

To measure the activity of moving vehicles in road traffic video, the lane locations should be determined first. Indeed the problem of lane finding has been studied for long [67] and [68]. It is a useful tool for road traffic monitoring systems to track vehicles appearing in the captured images. The lane locations can be determined with reference to the background road image [67], i.e. one that has no vehicle on it. However, for a real-time system, it is not possible to obtain a clean background road image before finding the lane locations. In [68], an activity map which accumulates all scene changes in a video is used to locate the lanes. The map can be used to distinguish between active areas of the scene where motion is found (the road) and inactive areas with insignificant motion (e.g. building). In this chapter, we adopt the approach in [68] and extend it to block-based to allow it to be compatible with the segmentation unit and the H.263 encoder. The lane finding process keeps track of the scene change information of all image blocks in the first

L seconds of the encoding process. The scene change information of a block can be obtained from the CDM of two consecutive frames. For every image block, the total amount of change as compared with that in the previous frame is evaluated. The activity map $A_{i,j}$ after L frames of lane finding interval is defined as follows:

$$A_{i,j} = \begin{cases} 1 & \text{if } \frac{1}{L}\sum_{n=1}^{L} d(i,j,n) \geq \psi_{scene} \\ -1 & \text{otherwise} \end{cases} \tag{4.1}$$

where $d(i,j,n)$ is defined as the amount of scene change of image block (i,j) in frame n from frame $n-1$; ψ_{scene} is a threshold value. Hence an image block (i,j) will be classified as a part of a lane if $A_{i,j}$ is equal to 1 or else if it has a value of -1. Figure 4.3 shows the lane finding algorithm.

4.4 *Motion analysis in video*

4.4.1 *Analysis based on the spatial correlation*

It is interesting to note that, in normal road traffic situation, the amount of scene change occurred in an image block of a lane can be highly correlated with its neighboring blocks along the same lane within a fixed time interval. This phenomenon exists because, if the lane is non-congested, vehicles on the same lane are normally traveling at similar speed and direction. If an image block of a non-congested lane is recorded with a particular scene change pattern across time, a similar pattern is expected for the next image block in the moving direction of vehicles, although with a time delay. Figure 4.4 further illustrates this idea. In Figures 4.4a and 4.4b, the amount of scene change in different frames of two neighboring image blocks of a non-congested lane is shown. They

demonstrate that the patterns of scene change across time for these two neighboring blocks are very similar, hence they have a high correlation.

However, it is not the case if the vehicles have unsteady speed, for instance, the vehicles are traveling on a congested lane. It is unsure if a vehicle will cause similar scene change to the next image block at the next instant of time. When measuring the correlation of the scene change pattern across time of two neighboring blocks along the moving direction of vehicles, a low value is expected. Our experiments have confirmed this idea. In Figure 4.5, we show the amount of scene changes in different frames of two neighboring blocks along the same congested lane. As the vehicles move with unsteady speed during queuing, the scene change patterns recorded in these two image blocks have different shape. Hence they have a low correlation value when comparing with each other.

The exact implementation method of traffic scene analysis using correlation is summarized as follows. First, we obtain from the lane finding process the location of lanes in the video. The lane finding process will run for a time interval of L frames and repeat after a fixed amount of time to avoid misdetection due to any sudden movement of the camera. After the lane finding process, the amount of scene changes for each image block is recorded within a time interval of K frames. Again, this process will also repeat after a fixed amount of time to avoid misdetection due to the change of traffic condition. Based on the recorded amount of scene changes in that K frames, the normalized cyclic correlation, as shown in Equation 4.2, is applied to every block and its neighboring block in the direction of vehicle motion to measure their similarity. The correlation values between the current motion block and surrounding blocks provide a qualitative description

of the traffic scene. In Equation 4.2, $d(i,j,n)$ is the total scene change of image block (i, j) in frame n from frame n-1.

$$R_{i,j}(x,y,c) = \frac{r_{i,j}(x,y,c)}{\frac{1}{k}\left[\sum_{n=1}^{k-1} d^2(i,j,n) \times \sum_{n=1}^{k-1} d^2(i+x, j+y, n)\right]^{1/2}} \tag{4.2}$$

where

$$r_{i,j}(x,y,c) = \frac{1}{k}\sum_{n=1}^{k-1} d(i,j,n)d(i+x, j+y, n+c);\ c = 0,1,\ldots,k\text{-}1;\ x \text{ and } y = -1, 0 \text{ or } 1.$$

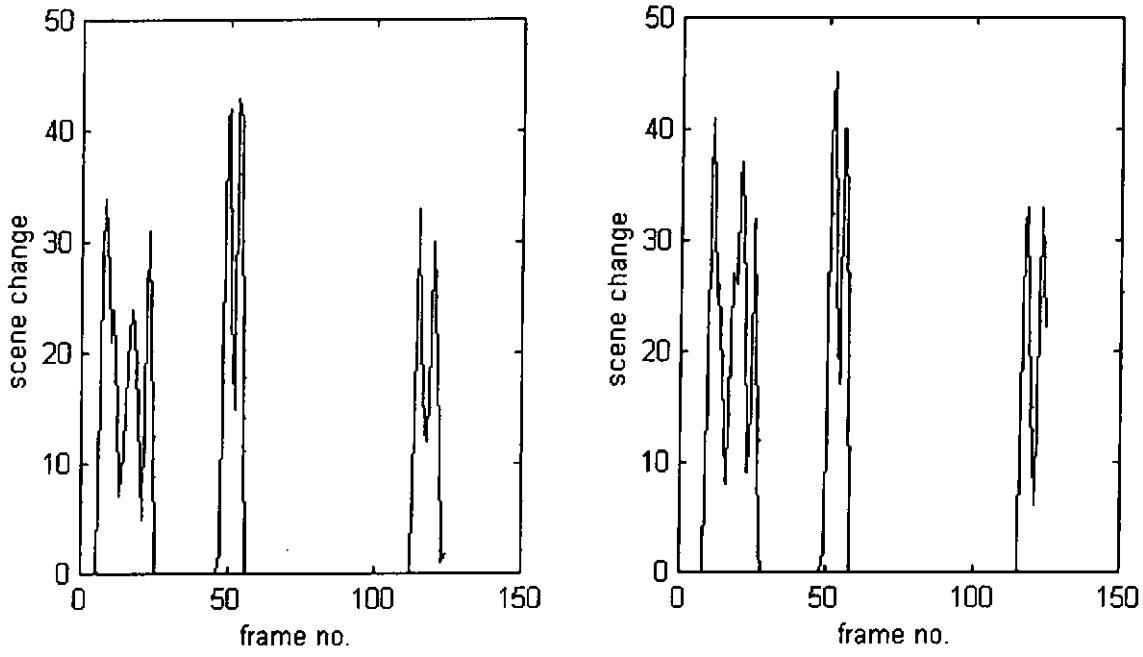
In Equation 4.2, the value of the indices x and y is -1, 0 or 1. It represents that the neighboring block can be at any one of the eight directions of the current block depending on the direction of vehicle motion. It is noted that the direction of vehicle motion can be obtained from the motion vectors generated by the video encoder. After $R_{i,j}(x,y,c)$ is obtained, the difference between maximum and minimum values of $R_{i,j}(x,y,c)$ is evaluated and compared with a threshold T_{corr} as in Equation 4.3:

$$M_{i,j} = \begin{cases} 1 & if \quad \left(\underset{c}{Max} R_{i,j}(x,y,c) - \underset{c}{Min} R_{i,j}(x,y,c)\right) > T_{corr} \ \ and \ \ A_{i,j} > 0 \\ 0 & if \quad \left(\underset{c}{Max} R_{i,j}(x,y,c) - \underset{c}{Min} R_{i,j}(x,y,c)\right) \le T_{corr} \ \ and \ \ A_{i,j} > 0 \\ -1 & if \quad A_{i,j} \le 0 \end{cases} \tag{4.3}$$

where $M_{i,j}$ is the so-called enhanced activity map which indicates the image block (i,j) is on a normal lane if it has a value of 1, or on a congested lane if it has a value of 0, or on a static background if it has a value of -1. By using Equation 4.3, the dependence of the correlation value to the image intensity is further reduced.

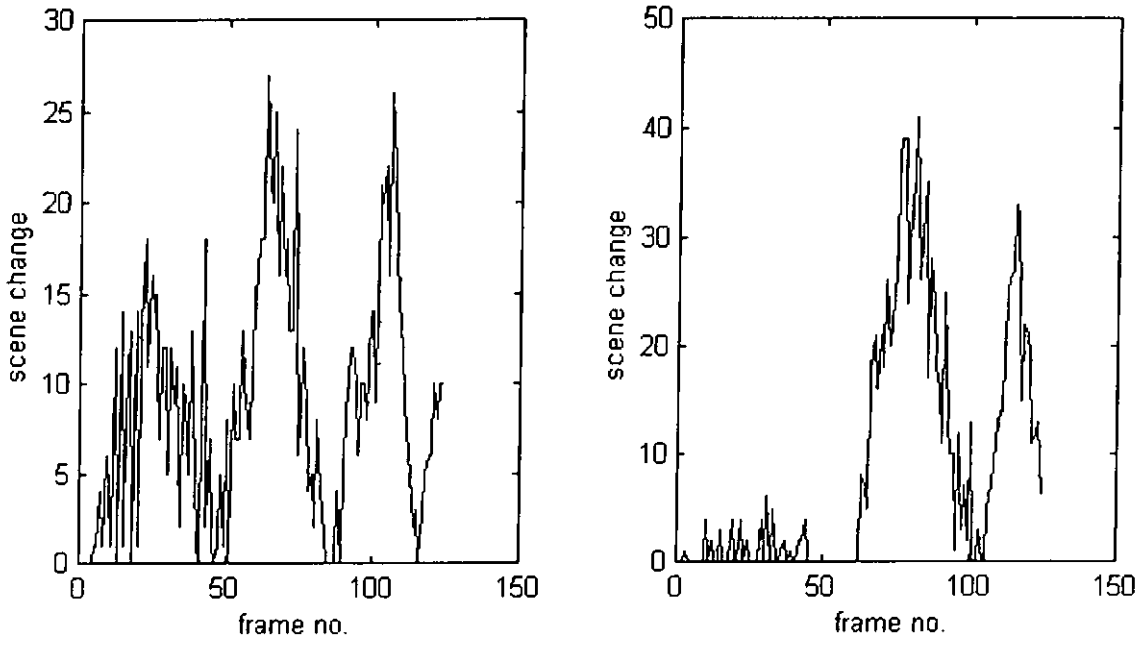
Figure 4.6 shows the testing video sequence, which is in QCIF format and has a frame rate of 30 frames/s. In Figure 4.6, there are 3 roads with vehicles moving at different speed. One road contains vehicles that are queuing in front of a tunnel (middle

part of Figure 4.6) and the others (bottom part and upper right part of Figure 4.6) contain vehicles with much higher speed. We first run the system by 10s for lane finding. For each of the lane obtained, we compute the correlation values based on the amount of scene change in the image blocks of that lane using Equations 4.2 and 4.3. The time interval for correlation computation is set to 10s with threshold T_{corr}=0.65. The correlation is taken with the pattern as shown in Figure 4.7. For each region, the average correlation value is computed as shown in Table 4.1.



(a) (b)

Fig. 4.4. (a) and (b) show the scene change of a block and its neighboring block along a non-congested lane.



(a) (b)

Fig. 4.5 (a) and (b) show the scene change of a block and its neighboring block along a congested lane.

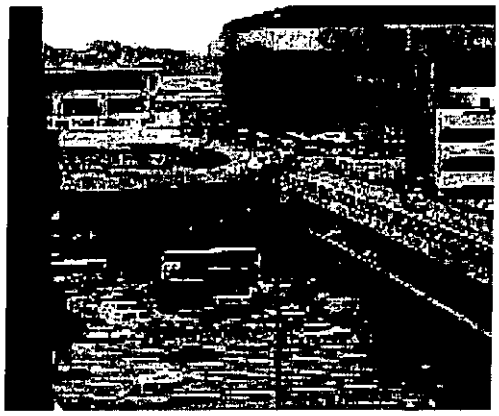


Fig. 4.6. Testing road traffic sequence.

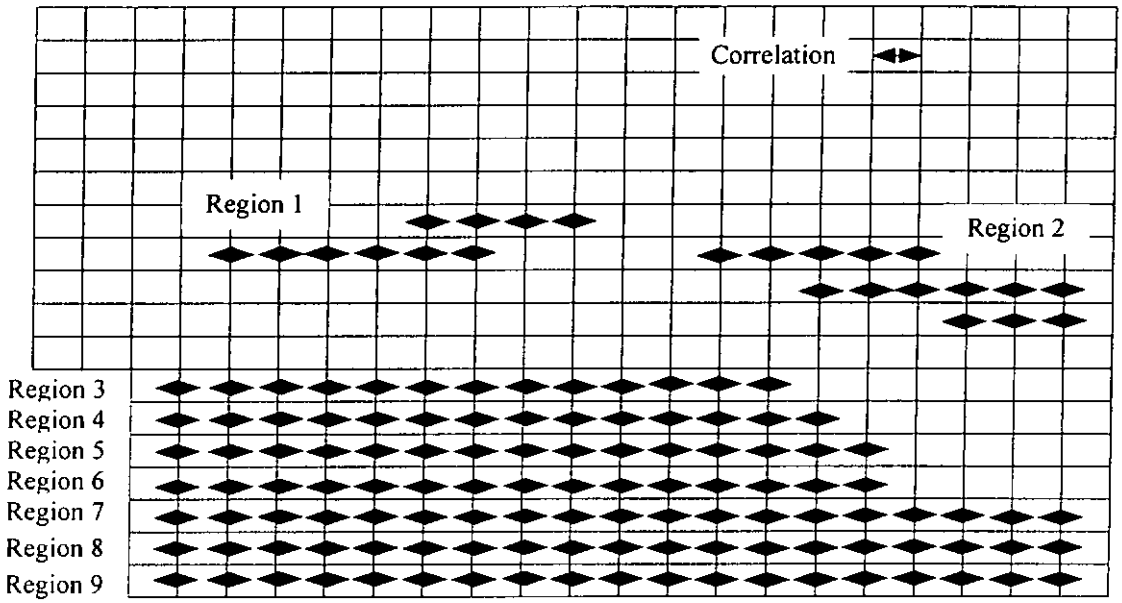


Fig. 4.7. Correlation pattern for the traffic video sequence. The spatial correlation is applied in each region. Each region is a lane that is described in the activity map. Within each region, correlation is measured on each pair of image blocks of the region.

Region	Average value
1	0.7965
2	0.8437
3	0.8541
4	0.5893
5	0.4637

6	0.8272
7	0.8409
8	0.9389
9	0.9144

Table 4.1. Average correlation value of each region in the video sequence.

As T_{corr} is set to 0.65, Table 4.1 indicates that region 4 and 5 should have traffic congestion. It matches perfectly with the actual traffic condition. Based on this information, we can optimally allocate resource for coding different regions, which is discussed in Section 4.5.1.

4.4.2 Analysis based on temporal information only

An essential information required by the correlation approach is the direction of vehicle motion. Based on this information, two adjacent blocks in the direction of vehicle motion are extracted and their correlation is measured. As mentioned earlier, the direction of vehicle motion can be obtained from the motion vectors generated by the video encoder. However, there are certain cases that this information is not feasible to obtain. For instance, when the scene is very complicated, motion vectors may be found in an irregular manner that it is difficult to conclude which motion vectors belong to which lanes. In this section, we propose an alternative approach that allows traffic analysis to be conducted in temporal domain only. It means that we do not require any kind of measurement between adjacent blocks but individually on each image block.

It is known that the scene change pattern of an image block of a non-congested lane should be more regular than that of a congested lane, since vehicles on a non-

congested lane often travel at steady speed. This idea has been illustrated in Figures 4.4 and 4.5 above. It can be verified from those figures that, for the image block of non-congested lane, its scene change pattern has less sharp change than that of congested lane. This observation implies that by measuring the density of sharp change, we can automatically distinguish a block is located in congested or non-congested region. A simple way to achieve this is by measuring the zero crossing density of the derivative of the scene change pattern of each image block across time. Here $d(i,j,n)$ is defined as the total scene change of image block (i, j) of frame n from frame n-1. Denote $Z(f)$ as the zero crossing representation of a function f such that $Z(f)$ is a binary sequence with every '1' in the sequence indicates the position of a zero crossing point in the function. The binary sequence of zero crossing for Figures 4.4a and 4.5a are shown in Figures 4.8 and 4.9 respectively. From these two graphs, it can be seen that the congested lane has higher density of zero crossing than non-congested lane as the scene change pattern of congested lane has more sharp change. Based on these parameters, the enhanced activity map $M_{i,j}$ as defined in Equation 4.3 is now redefined as follows:

$$M_{i,j} = \begin{cases} 1 & if \quad \frac{1}{N}\sum_{n \in N} Z\left(\frac{\partial d(i,j,n)}{\partial n}\right) \le T_{density} \quad and \quad A_{i,j} > 0 \\ 0 & if \quad \frac{1}{N}\sum_{n \in N} Z\left(\frac{\partial d(i,j,n)}{\partial n}\right) > T_{density} \quad and \quad A_{i,j} > 0 \\ -1 & if \quad A_{i,j} \le 0 \end{cases} \qquad (4.4)$$

where $A_{i,j}$ is the activity map defined in Equation 4.1; N is the total number of sampling frames and $T_{density}$ is the density threshold. Equation 4.4 shows that the enhanced activity map is now evaluated by using the average number of zero crossing occurred when differentiating the scene change pattern across time. As similar in Equation 4.3, $M_{i,j}$ indicates the image block (i, j) is on a normal lane if it has a value of 1, or on a congested lane if it has a value of 0, or on a static background if it has a value of -1. Table 4.2 shows

the zero crossing density of the sharp change for the blocks in video sequence shown in Figure 4.6. By setting $T_{density}$ equals 0.35, the non-congested and congested regions are separated. One of the advantages of the proposed approach is it ensures the estimation to be done irrespective to the magnitude of the input signal. Hence even a block contains only part of vehicles, which in turns diminishes the magnitude of the signal, the density of sharp changes in the signal can still be estimated.

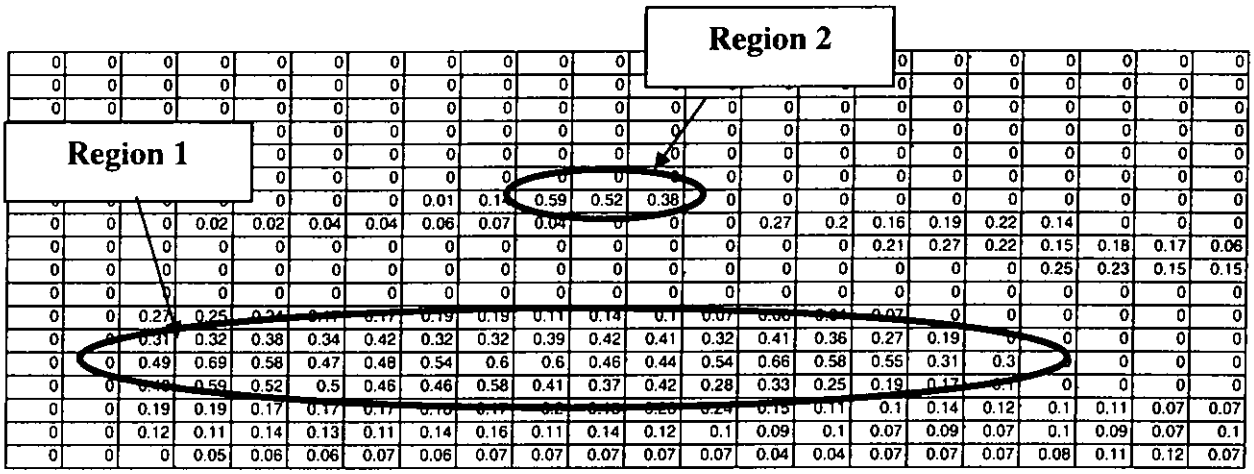


Table 4.2. The density of sharp change for the blocks in the video sequence.

Nevertheless, we notice that for some image blocks on the non-congested lane, they may exhibit a scene change pattern similar to that in the congested lane if they are heavily contaminated by noise (such as Region 2). Misdetection may occur in this case and introduce a slight decrease in the SNR of the reconstructed video; since the system will consider these blocks are on the congested lane and use a lower frame rate for their coding. It is interesting to note that this kind of misdetection does not exist in the result of the correlation approach. It is because the correlation approach targets at measuring the average similarity of the scene change pattern of neighboring blocks, which is less susceptible to noise. This concludes that the correlation approach will generally perform

better when the acquired video is noisy, while the zero crossing approach is more suitable when it is infeasible to obtain the prior information of the direction of vehicle motion.

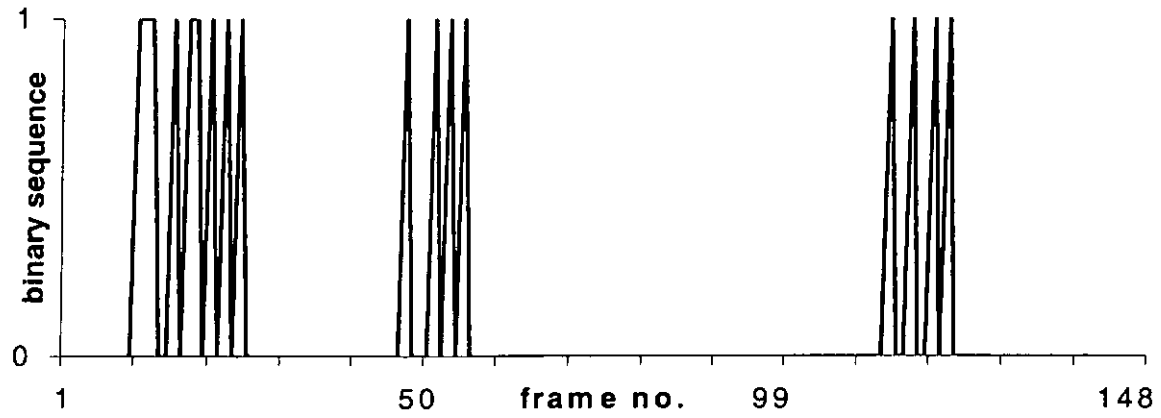


Fig. 4.8. Binary sequence for the zero crossing density of the derivative of the scene change pattern in Figure 4.4a.

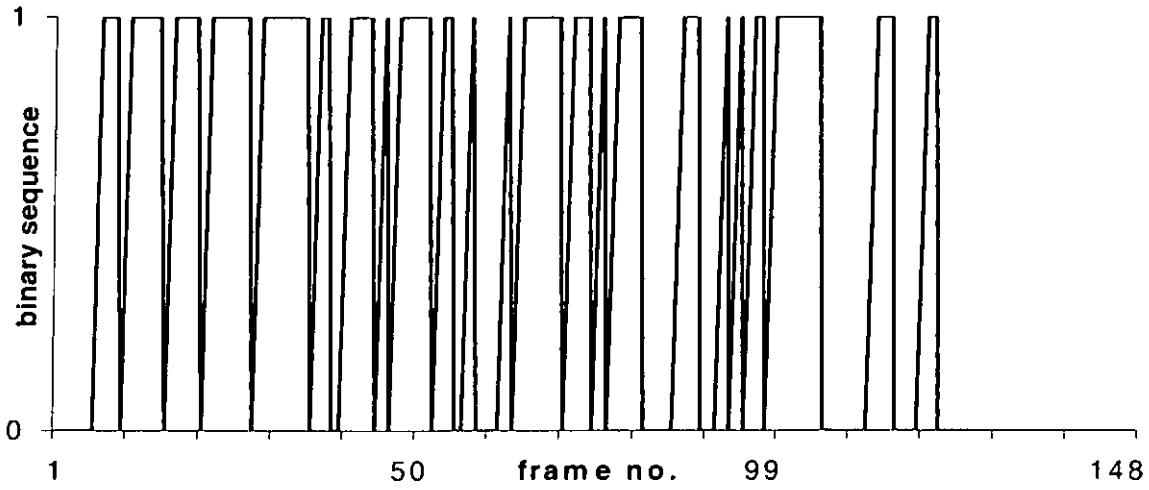


Fig. 4.9. Binary sequence for the zero crossing density of the derivative of the scene change pattern in Figure 4.5a.

4.5 *Removal of redundancy*

4.5.1 Adaptive temporal redundancy reduction

Once the location of lanes is found and the lanes are classified as congested or not, the next step is to make use of this information to control the coding frame rate of the moving objects on the lanes to reduce the temporal redundancy. An intuitive way to achieve this is to directly instruct the control unit of the H.263 encoder to code according to the congestion situation of the lane. This approach however may require a drastic modification of the control unit to implement the complicated control logic. In our method, the temporal redundancy reduction is achieved externally by simply adjusting the threshold Th_{mb} of the segmentation unit as shown in Figure 3.2 in Chapter 3. As it is mentioned above that the threshold Th_{mb} controls if an image block should be classified as a MB. If not, the coding of the image block will be skipped (by setting COD to 1) during the INTER frame encoding phase. Hence by adjusting the threshold Th_{mb}, we can control the coding rate of an image block.

Let us use an example to illustrate the whole mechanism of coding rate control. Assume that we obtain the information from the enhanced activity map, $M_{i,j}$, that an image block (i, j) is on a congested lane. When coding this image block, we set the threshold Th_{mb} to a higher value so that the image block cannot be classified as MB initially. In this case, the coding of this image block will be skipped during the INTER frame encoding phase. While the coding of this image block is skipped, the difference of it as compared with that in the motion compensation unit will accumulate. After a few frames, it will reach to a point that the difference is so big that exceeds the threshold Th_{mb}. The image

block will then be classified as MB and coded in the normal manner. The image block in the motion compensation unit will then be updated. It is seen that by using the threshold Th_{mb}, the coding rate of an image block can be easily adjusted.

4.5.2 Spatial redundancy reduction

The sensitivity of human visual system varies greatly depending on the spatial and temporal frequency of the viewing image [69]. These variations can be exploited to determine how the image information can be discarded without subjectively degrading the final image. As indicated in [69], human eyes have the characteristic of band-pass filtering on the spatial frequency axis. Depending on the speed of the moving objects in the image, the peak frequency and passband are different. As the velocity of the object increases, the peak frequency approaches zero and the relative sensitivity decreases [69]. Hence a coarser processing is permitted in the parts where the movement of the object is high.

Many methods have been suggested in literature to remove subjective redundancy using human visual system (HVS) characteristics. Subband coding using HVS [70] and [71] is based on the relative sensitivity of human vision in each spatial frequency. Each image frame is decomposed into several subbands and coding for each subband is based on the perceived resolution loss as a result of movement in scenes [71] or in some analysis-synthesis systems [70]. Other characteristics, such as luminance dependence [72] and masking effects [72] and [73] have also been used in video coding. A simple and efficient method of removing subjectively redundant information is suggested in [74]. The image transform coefficients are filtered using the psychovisual thresholding technique and followed by the process of psychovisual quantization. In our approach, only the

psychovisual thresholding method is adopted because of the real time purpose of the encoder. Since the psychovisual thresholding technique is based on the sensitivity of the HVS to different spatial frequencies, we can simply implement it by comparing the DCT coefficients with a psychovisual threshold, as shown in Figure 4.1. Hence, no modification to the format of the encoded bit stream is required as opposed to the psychovisual quantization. For the evaluation of the threshold value, it is noted that the threshold is inherently adaptive to the changing image characteristic since it changes with the DC transform coefficient C_{00}. More specifically, the threshold level matrix $S'_{i,j}$ for the 8 x 8 DCT coefficients, as shown in Table 4.3, is computed as follows:

$$S'_{i,j} = A\left(\frac{S_{i,j}}{P_{i,j}}\right)^{-1} \tag{4.5}$$

and the psychovisual threshold value for each coefficient $T_{i,j}$ is then given as:

$$T_{i,j} = S'_{i,j} \times C_{00} \tag{4.6}$$

where $S_{i,j}$ is the sensitivity function and $P_{i,j}$ is the power distribution function given in [74]. i, j are the indices to the image and A is a scaling factor.

0	1	1	1	1	2	3	5
1	1	1	1	1	2	3	5
1	1	1	1	2	3	4	7
1	1	1	2	3	4	6	9
2	2	3	3	4	6	9	13
4	4	5	6	8	10	14	20
9	9	10	12	14	18	23	28
17	17	19	21	24	28	30	28

Table 4.3. Threshold level matrix, $S'_{i,j}$.

As mentioned before, a coarser encoding process is permitted in the regions where the movement of the objects is high. Hence this psychovisual threshold is applied only to

the image blocks with high mobility. The mobility information is obtained from the enhanced activity map. Based on the map, the motion blocks that are located on non-congested lanes are coarsely quantized by applying the psychovisual threshold to remove the spatial redundancy. For the other image blocks, the psychovisual threshold will not be applied in order to prevent the degradation of image quality.

4.6 Results and discussions

We implemented the proposed system with a 450MHz Pentium II personal computer. The traffic condition of two roads in two different geographical locations is captured and saved as two video sequences for testing. Both sequences are in the QCIF format.

We first run the system in the pre-processing stage, which includes lane finding and traffic analysis, for 20s then encoded the road traffic video based on the enhanced activity map of the image blocks. Both traffic video sequences were coded with a fixed quantization step-size and frame rate as shown in Tables 4.4 and 4.5. In Tables 4.4 and 4.5, congestion detection using the correlation approach and the zero crossing density detection approach are compared. We also compare the bit rate of the encoded bit stream generated by the H.263 encoder with and without the proposed temporal and spatial redundancy reduction techniques. In the tables, average peak signal to noise ratio (PSNR) of the reconstructed frames is used as a distortion measure and is given by:

$$Average\ PSNR = 10\log\frac{255^2}{\frac{1}{N}\sum_{i=1}^{N}(o_i - r_i)^2} \tag{4.7}$$

where N is the number of samples and o_i and r_i are the amplitudes of the original and reconstructed pixels, respectively. In Tables 4.4 and 4.5, the term "Bit rate" is the average bit rate for the encoded bit stream and "bpp" is the number of bits per pixel of a video frame. From Table 4.4, it is seen that if the zero crossing density detection approach is adopted for congestion detection, there is a decrease of 30% in bit rate while the average PSNR just decreases by 0.5 dB as compared with the original H.263 encoder. When comparing between the approach of correlation and zero crossing density detection, the average compression ratio of using the zero crossing density detection approach is generally higher but keeping the PSNR unchanged. This indicates that, while the video sequence is not that noisy, the zero crossing density approach gives a better classification performance than the correlation approach due to the complexity of the video scene. Table 4.5 shows the result of encoding another road traffic video sequence. When comparing with the original H.263 encoder, there is a decrease of 19% in bit rate while the average PSNR just decreases by 0.8dB. To illustrate particularly the improvement due to congestion detection, we compare the current approach with one of our previously proposed approaches of which only the foreground-background segmentation is used but without congestion detection [66]. It is seen in Tables 4.4 and 4.5 that a maximum of 10% decrease in bit rate is achieved with a negligible distortion. Figures 4.10 and 4.11 show two sets of samples of the decoded frames for the two road traffic video sequences. The corresponding compression ratios (bpp) for each video frame are plotted in Figures 4.12 and 4.13. It is seen that, as compared with the original H.263 codec, the decoded images of the proposed system give negligible visual distortion. However, more than 20% reduction of the transmission data can be achieved with the proposed system.

Sequence	Traffic1			
Mode	Proposed system (correlation approach is used in pre-processing step)	Proposed system (zero crossing density detection approach is used in pre-processing step)	Proposed system with no congestion detection [66]	H.263 [5]
Format	QCIF			
Quantization	15			
Frame rate	30Hz			
PSNR[dB]	32.4	32.4	32.5	32.9
Bit rate[kbps]	59.6	58.5	65.7	83.6
Average bpp	0.0780	0.0774	0.0859	0.1090

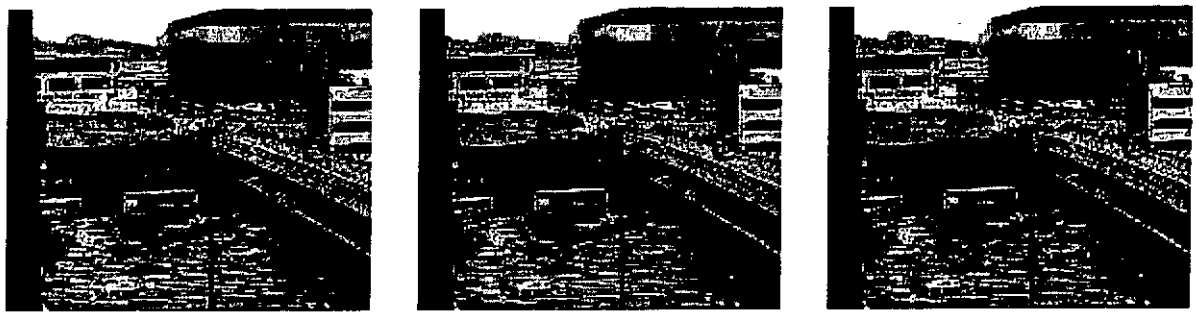
Parameters: Th_{ch} = 200, Th_{mb} = 5 (for fast object) and 10 (for other), ψ_{scene} = 0.25, T_{corr} = 0.65, $T_{density}$ = 0.35.

Table 4.4. Results of road traffic video sequence 1.

Sequence	Traffic2			
Mode	Proposed system (correlation approach is used in pre-processing step)	Proposed system (zero crossing density detection approach is used in pre-processing step)	Proposed system with no congestion detection [66]	H.263 [5]
Format	QCIF			
Quantization	15			
Frame rate	20Hz			
PSNR[dB]	30.5	30.5	30.6	31.3
Bit rate[kbps]	49.1	49.1	51.8	60.5
Average bpp	0.0966	0.0964	0.1018	0.1189

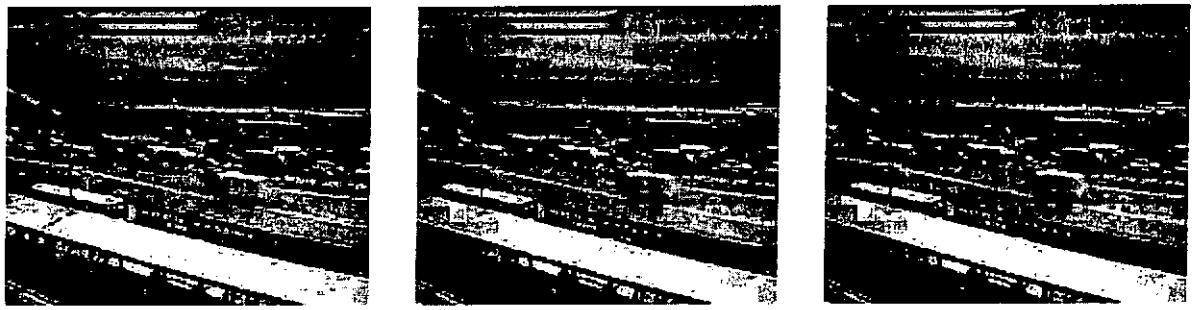
Parameters: Th_{ch} = 100, Th_{mb} = 5 (for fast object) and 10 (for other), ψ_{scene} = 0.25, T_{corr} = 0.65, $T_{density}$ = 0.34.

Table 4.5. Results of road traffic video sequence 2.



(a) (b) (c)

Fig. 4.10. (a) The decoded frame (frame 10) of the original H.263 bit stream (traffic sequence 1). (b) and (c) The same decoded frame that is encoded by the proposed system using the zero crossing density detection and correlation approaches in congestion detection, respectively.



(a) (b) (c)

Fig. 4.11. (a) The decoded frame (frame 100) of the original H.263 bit stream (traffic sequence 2). (b) and (c) The same decoded frame that is encoded by the proposed system using the zero crossing density detection and correlation approaches in congestion detection, respectively.

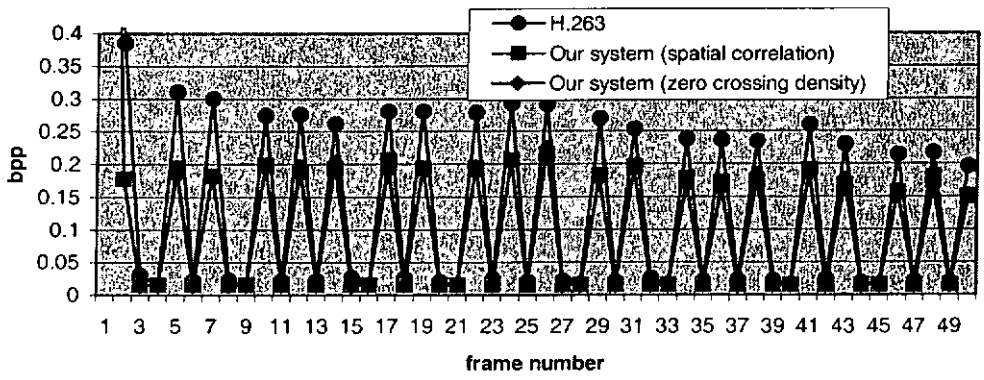


Fig. 4.12. A plot of bpp against frame number for Traffic 1 sequence.

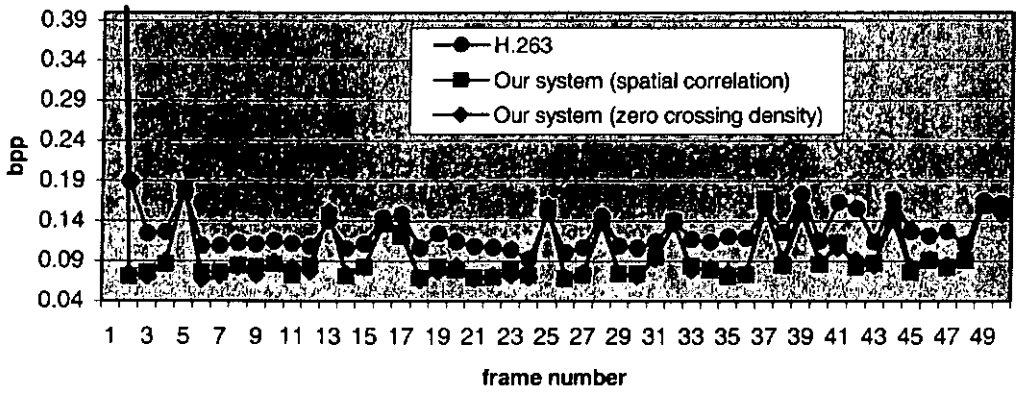


Fig. 4.13. A plot of bpp against frame number for Traffic 2 sequence.

4.7 Summary

In this chapter, we present a content-based scalable H.263 video coding system that is suitable for road traffic monitoring. It has been shown that the bit rate of the proposed system decreases as compared with the conventional H.263 codec. The decrease in bit rate can be more than 20%. This bit rate is suitable for video transmission over current mobile data packet network, such as, GPRS. Besides, the encoded video sequence can be decoded with conventional H.263 decoder, which is different from other proprietary codec that requires a tailor-made decoder. Besides road traffic monitoring, the proposed system can also be applied to other video surveillance systems with fixed camera setting.

Chapter 5

Temporally scalable video coding using interpolating wavelet transform

5.1 Introduction

Scalability has been one of the most important features of modern video coding systems. For video communication systems, a scalable video codec allows adaptation of video quality to the bandwidth of communication channels. For storage and retrieval systems, it enables fast retrieval and hence enhances the flexibility of the retrieval systems. Among the different techniques available, temporal scalability allows extraction of video of multiple frame rates from a single coded stream. It can, in general, work with other scalable video coding techniques, such as resolution and SNR scalable coding, to enhance the degree of scalability.

In recent years, various temporally scalable video coding algorithms have been proposed [75], [32]. They can be divided into two categories: motion compensated prediction (MCP) coding [75], [39] and temporal subband (TSB) coding [76], [77], [32]. Traditional MCP schemes do not allow the implementation of temporal scalability. It is made possible by introducing an extra amount of reference frames in each group of frames.

TSB approaches on the other hand implement the idea of temporal scalability by using the multiresolution characteristic of subband coding in the temporal domain. It was reported [78] that the performances of MCP and TSB are similar for full frame rate video. For lower frame rates, MCP is superior to TSB. Nevertheless, TSB is often much simpler than MCP due to the exclusion of the complicated motion compensation process. There are studies which try to improve the performance of TSB by incorporating this process. The resulting coding algorithm is known as MC-TSB [45], [49], [79]. Although MC-TSB sacrifices the simplicity of the TSB coding algorithm, it is still inferior to MCP in generating lower frame rate videos [78].

One of the major reasons leading to the poor performance of TSB is that the lower frame rate videos are obtained from the low-pass subbands of the video codec. While the low-pass subbands in general give the moving average of the video frames temporally, blurred motion often results and leads to a decrease in image quality. Another problem of TSB is due to the floating point data generated by the subband filters. It is known that the filter coefficients of most of the subband filters are real numbers. The filtered video data will inevitably be real numbers also. While the original video data are often represented by integers, the floating-point filtered video implies an increase of resolution and hence entropy. It also introduces a difficulty if we want to extend the video codec to lossless mode, which is desirable in many medical applications. Although the problem can be solved by scaling up the filter coefficients to integer, the variance of the scaled filter coefficients will increase and the efficiency of subsequent spatial coding of the filtered data will drop accordingly.

In this chapter, a new TSB based video coding algorithm is proposed. It not only shares the same advantage of TSB in that its architecture is very simple, but also resolves the problems of TSB by (i) adopting the interpolating wavelet basis [46] to replace the traditional Haar wavelet basis; (ii) using the new reversible rounding method to convert the coefficients generated by the interpolating wavelets to integers. By using the interpolating wavelet bases, the lower frame rate video is no longer the blurred version of the original video, but is composed by exactly the original video frames. With the proposed reversible rounding method, the transform coefficients generated by the interpolating wavelets are converted to integer form. The reduced resolution due to conversion will be compensated on the decoder side hence a perfect reconstruction is achieved. This feature also allows the new algorithm to be applicable to lossless video coding.

We compare the performance of the proposed algorithm with traditional TSB approaches that make use of the Haar wavelet basis. The performances of the proposed algorithm with and without the reversible rounding method are also compared. To complete the comparison, we use JPEG2000 [11] image codec for the coding of each temporally wavelet-transformed video frame. Experimental results show that the proposed algorithm using interpolating wavelet transform and the reversible rounding method outperforms the traditional one both objectively and subjectively. The algorithm is particularly suitable to video storage and retrieval systems in which the temporal scalability feature of the new codec allows fast scanning of the contents in the stored video. It can be extended to lossless video coding. For this kind of applications, we compare the performance of the proposed coding algorithm with the Motion-JPEG2000 [10] and the

results show that a higher compression ratio and the same temporal scalability can be achieved.

This chapter is organized as follows. In Section 5.2, a temporally scalable video coding algorithm by using the interpolating wavelet basis is proposed. In Section 5.3, the reversible rounding method is introduced. In Section 5.4, the temporal scalability of the proposed coding algorithm is discussed. Comparison results are given in Section 5.5. The chapter is concluded in Section 5.6.

5.2 Video coding based on interpolating wavelets

In Chapter 2, we have defined the interpolating wavelet transform. We have indicated that Donoho's interpolating wavelets are non-orthogonal and have less degree of freedom for filter optimization. Orthogonal and biorthogonal interpolating wavelets were suggested recently to solve these problems [80], [81]. However, in our method, the original interpolating wavelet transform is considered due to its extremely short analysis filters. For example, a possible multiresolution analysis based on Donoho's interpolating scaling and wavelet functions is shown in Figure 5.1:

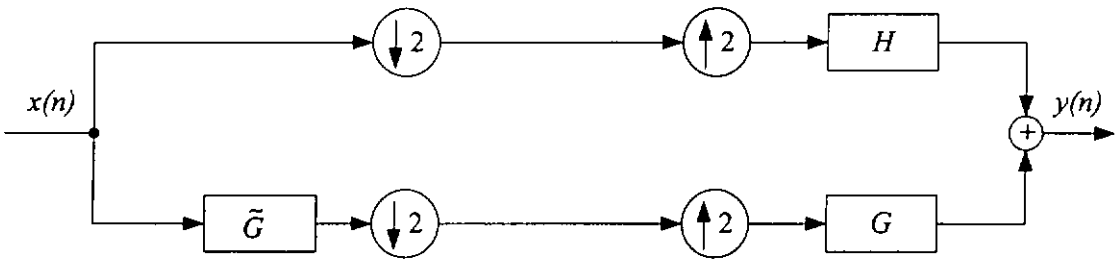


Fig. 5.1. Multiresolution analysis based on interpolating scaling and wavelet functions.

where the analysis filter $\tilde{H}$ is a delta function (hence is not shown in the diagram); the other filters $\tilde{G}$, G and H are as follows:

$$\tilde{G}(z) = -0.5z^2 + z - 0.5 \tag{5.1}$$

$$G(z) = z^{-1} \tag{5.2}$$

$$H(z) = 0.5z + 1 + 0.5z^{-1} \tag{5.3}$$

When applying the interpolating wavelets to video coding applications, short filter length is extremely important since it reduces the memory requirement for frame buffer. For video with drastic motion, short filter also avoids the generation of too many high frequency data due to the low correlation between frames.

The proposed video codec comprises a separated interpolating wavelet transform for the coding of video data temporally and followed by a JPEG2000 encoder to encode each transformed frame. Temporal decomposition is performed using Deslauriers-Dubuc filter [47] with 3 taps as shown in Figure 5.1 and Equations 5.1 – 5.3. Unlike the traditional TSB approaches using the Haar wavelet basis, the interpolating wavelet transform removes temporal redundancy between frames in the high-pass branch and bypass half of the video frames in the low-pass branch. Hence the video given in the low-

pass subband is composed by the original frames with a frame rate half of the original one. This is the major difference from traditional TSB approaches in which the lower rate video is the moving average of video frames.

For the proposed coding system, each transformed video frame generated by the low-pass subband is encoded using the JPEG2000 encoder. The coefficients of the high-pass subband are fixed to integer by combining with the output generated by the so-called *S* unit, in which the reversible rounding method is performed. Figure 5.2 illustrates the new video coding system. Figure 5.3 shows the internal structure of the S unit.

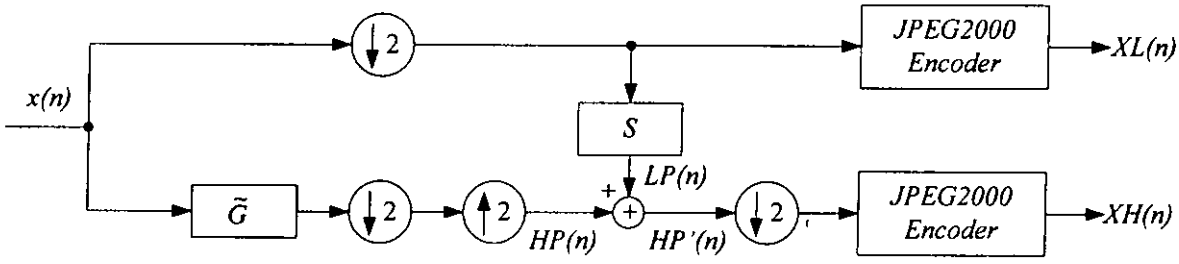


Fig. 5.2. Temporally scalable video encoder based on interpolating wavelet transform.

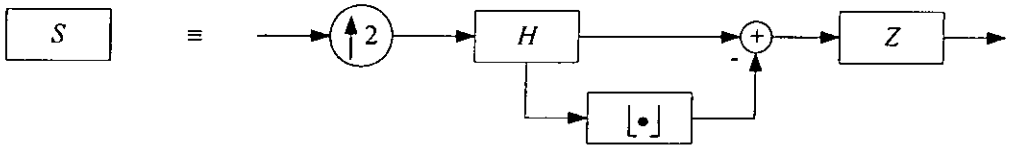


Fig. 5.3. The internal structure of the S unit. The symbol $\lfloor . \rfloor$ stands for rounding to the nearest smaller integer.

5.3 *Converting transform coefficients to integers*

The output *HP(n)* (as shown in Figure 5.2) generated from the high-pass branch contains in general moving edges of the video frames while removing all stationary areas. Therefore these frames contain less texture than their original frames and hence fewer bits will be generated when they are applied to the subsequent spatial encoder. However, since most of the wavelet filters (including the Deslauriers-Dubuc filters or Haar filters) have floating point coefficients, *HP(n)* which is generated by these filters will inevitably be floating point numbers. While the original video data are often in integer format, the increase of resolution in representing *HP(n)* implies an increase in entropy and hence lowers the efficiency of subsequent spatial coding. The existence of floating point data also disables the extension of video codec to lossless coding, which is desirable in many medical applications. It is because the floating point data in *HP(n)* cannot be exactly reconstructed even if a lossless spatial codec, such as JPEG2000 (lossless mode), is used. Although *HP(n)* can be scaled up to integer, the variance of the scaled *HP(n)* will increase and the efficiency of the subsequent spatial coding will decrease as is the case for lossy mode.

To convert the floating point data to integers, we propose a reversible rounding method. From Figure 5.1, $\tilde{H}(z)$ and $G(z)$ can be represented in z domain as follows:

$$\tilde{H}(z) = 1 \text{ and } G(z) = z^{-1}$$

We rewrite $\tilde{G}(z)$ and $H(z)$ in z domain as follows:

$$H(z) = H_i(z) + H_d(z) \tag{5.4}$$

and $$\tilde{G}(z) = \tilde{G}_i(z) + \tilde{G}_d(z) \tag{5.5}$$

where $H_i(z)$ and $\tilde{G}_i(z)$ are integer portions and $H_d(z)$ and $\tilde{G}_d(z)$ are decimal portions of $H(z)$ and $\tilde{G}(z)$ respectively. To achieve perfect reconstruction, it requires that,

$$H(z) + z^{-1}\tilde{G}(z) = 1 \tag{5.6}$$

which implies that

$$H_d(z) + z^{-1}\tilde{G}_d(z) \quad \text{must be integers.} \tag{5.7}$$

According to Figures 5.2 and 5.3,

$$LP(z) = \frac{1}{2}[X(z) + X(-z)] \cdot H_d(z) \cdot z \tag{5.8}$$

$$\text{and } HP(z) = \frac{1}{2}\left\{X(z)\left[\tilde{G}_i(z) + \tilde{G}_d(z)\right] + X(-z)\left[\tilde{G}_i(-z) + \tilde{G}_d(-z)\right]\right\}. \tag{5.9}$$

Hence,

$$HP'(z) = \frac{1}{2}\left\{X(z)\left[zH_d(z) + \tilde{G}_i(z) + \tilde{G}_d(z)\right] + X(-z)\left[zH_d(z) + \tilde{G}_i(-z) + \tilde{G}_d(-z)\right]\right\}$$

That is,

$$HP'(z) = \frac{1}{2}\begin{Bmatrix} X(z)\tilde{G}_i(z) + X(-z)\tilde{G}_i(-z) + \\ X(z)\left[zH_d(z) + \tilde{G}_d(z)\right] + X(-z)\left[zH_d(z) + \tilde{G}_d(-z)\right] \end{Bmatrix} \tag{5.10}$$

Equation 5.10 shows that *HP'(z)* has integer coefficients. It can be proven as follows. First the upper part of Equation 5.10 must be integers since both *X(z)* and $\tilde{G}_i(z)$ are integers. Let us consider the lower part of Equation 5.10. From Equation 5.7, $zH_d(z) + \tilde{G}_d(z)$ must be integers and is equal to either 0 or 1. For Deslauriers-Dubuc filters, $H(2n) = 0$ if $n \neq 0$, hence $zH_d(z) = -zH_d(-z)$. Together with Equation 5.7, it is known that $zH_d(z) + \tilde{G}_d(-z)$ must also be integers and is equal to either 0 or 1. Hence the lower part of Equation 5.10 must also be integers. That is, *HP'(z)* must have integer coefficients.

It is seen that since $zH_d(z)+\tilde{G}_d(z)$ and $-zH_d(-z)+\tilde{G}_d(-z)$ are either 0 or 1, it represents some minor adjustments to *HP(n)*. These minor adjustments allow *HP'(n)* and hence *XH(n)* (as shown in Figure 5.2) to be integers. This process can be reversed on the decoder side as shown in Figure 5.4.

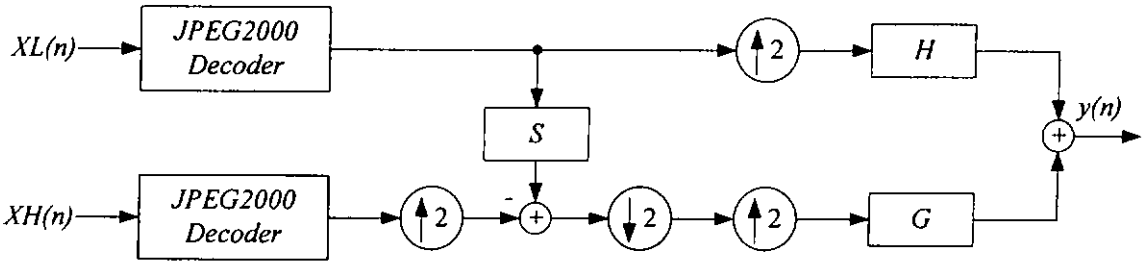


Fig. 5.4. Temporally scalable video decoder with the S unit.

It is seen in Figure 5.4 that the minor adjustments are subtracted from *XH(n)* to obtain the original high pass video frames. If the JPEG2000 codecs are lossless, *y(n)* will exactly be equal to *x(n)* due to the perfect reconstruction property of the wavelet transform. Hence the proposed reversible rounding method extends the lossless image codecs to video coding applications. However, it is not necessary for the spatial codec to be lossless when applying this method. Our experiments show that it is also useful for lossy spatial coding in improving the rate distortion performance.

5.4 Temporal scalability by interpolating wavelets

As seen in Figure 5.2, *XL(n)* contains half of the original input images and is encoded directly using the JPEG2000 encoder. The encoded bit stream can be decoded

independently without considering the results in the high pass branch. We can therefore achieve temporal scalability with the proposed coding algorithm. For instance, assume there is a video communication system that is serving for clients with different channel bandwidth. For clients with high channel bandwidth, both *XL(n)* and *XH(n)* are sent. For clients with low channel bandwidth, only *XL(n)* is sent. For these clients, the bit stream can be decoded correctly on the decoder side but the frame rate is dropped by two. We can introduce a higher degree of scalability by increasing the decomposition level as shown in Figure 5.5.

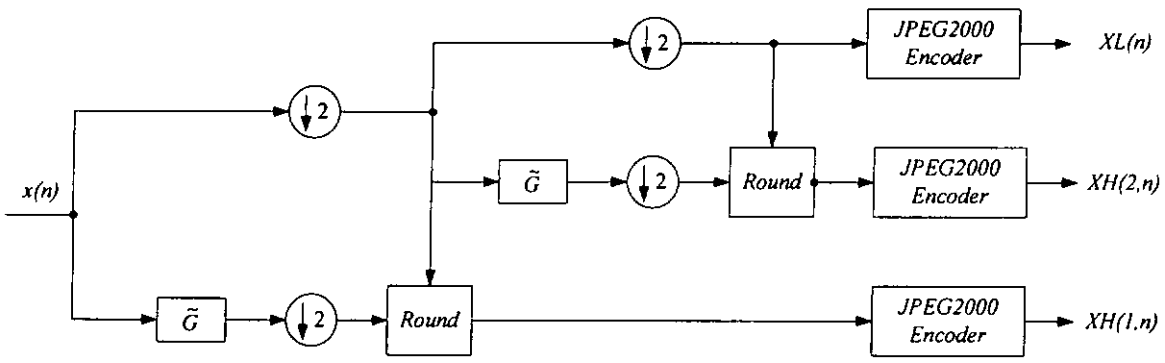


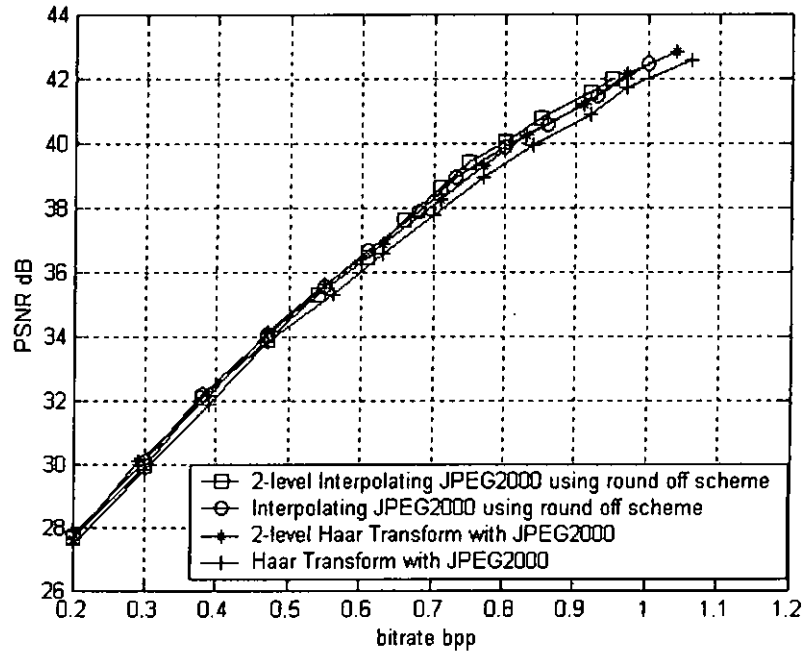
Fig. 5.5. Temporally scalable video coding with decomposition level equals to 2.

If only *XL(n)* is transmitted, a frame rate of *F*/4 can be achieved on the decoder side. When *XH(2,n)* is also transmitted, a frame rate of *F*/2 can be achieved. If the decoder also receives *XH(1,n)*, a full frame rate video can be displayed.

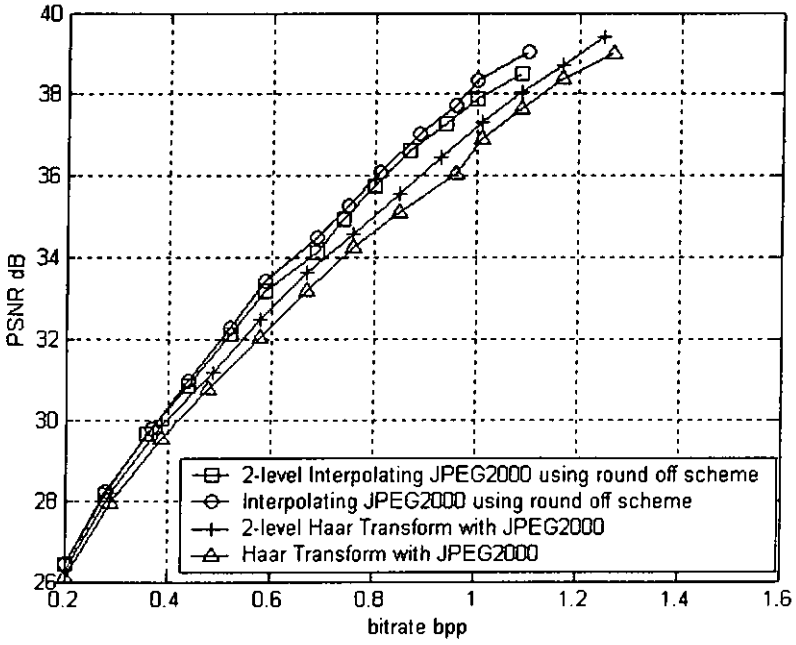
5.5 *Experimental results*

A simulation of the proposed temporally scalable video coding scheme was performed and the rate-distortion performance is evaluated on three standard gray level video sequences, "*Carphone*", "*Foreman*" and "*Claire*". Each video sequence has 64 frames with a resolution of 176x144. The performances of the proposed coding scheme working in both lossy and lossless compression modes are tested. They are compared with the traditional TSB approach in the lossy case and Motion JPEG2000 in the lossless case.

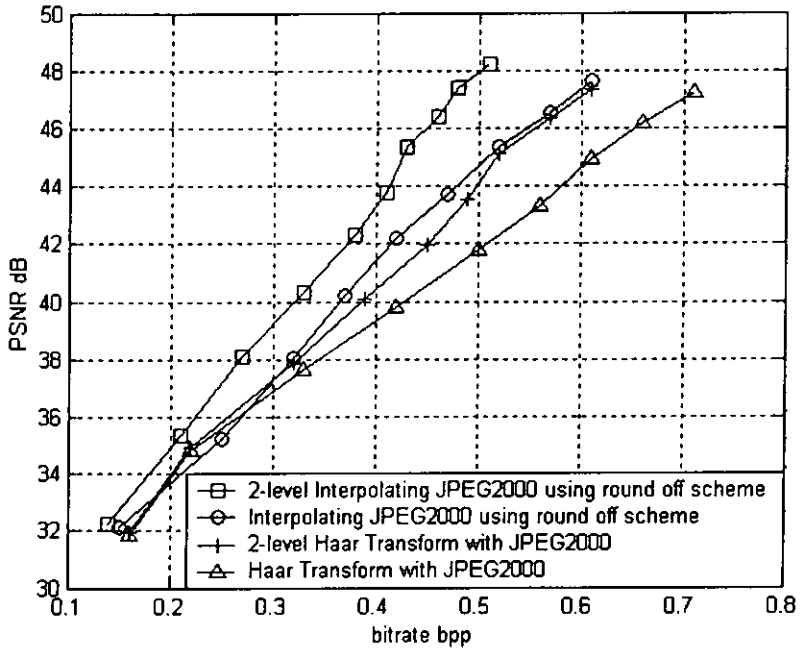
Figures 5.6 – 5.9 show the results when comparing with the traditional TSB approach in lossy compression mode at different frame rates. It is seen in Figure 5.6 that, at full frame rate, the performance of the proposed approach is similar, if not better, than the traditional TSB approach using the Haar wavelet basis. It performs particularly well for video with less motion, such as, *Claire*. Figure 5.6 also shows that their performance is better when the number of decomposition level is larger.


PSNR dB
bitrate bpp
2-level Interpolating JPEG2000 using round off scheme
Interpolating JPEG2000 using round off scheme
2-level Haar Transform with JPEG2000
Haar Transform with JPEG2000

(a)

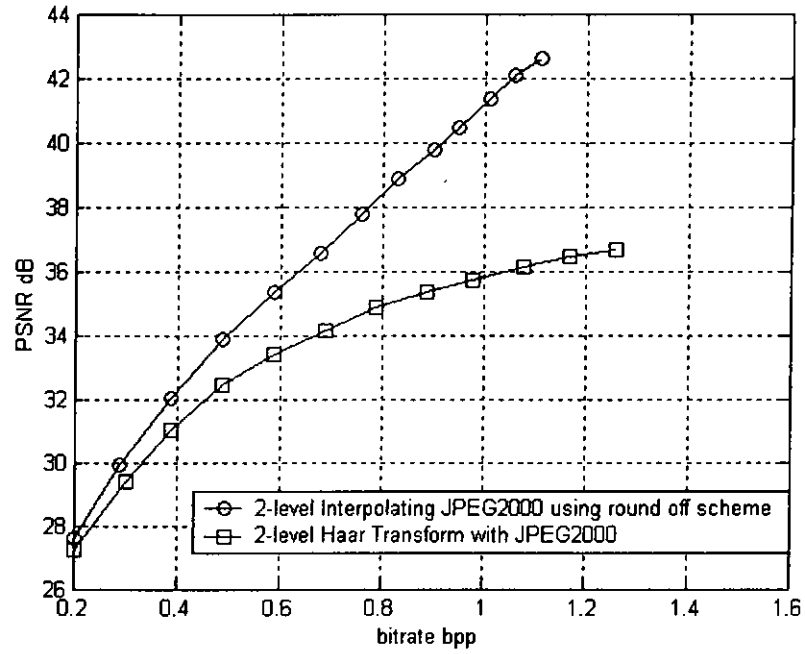

PSNR dB
bitrate bpp
2-level Interpolating JPEG2000 using round off scheme
Interpolating JPEG2000 using round off scheme
2-level Haar Transform with JPEG2000
Haar Transform with JPEG2000

(b)

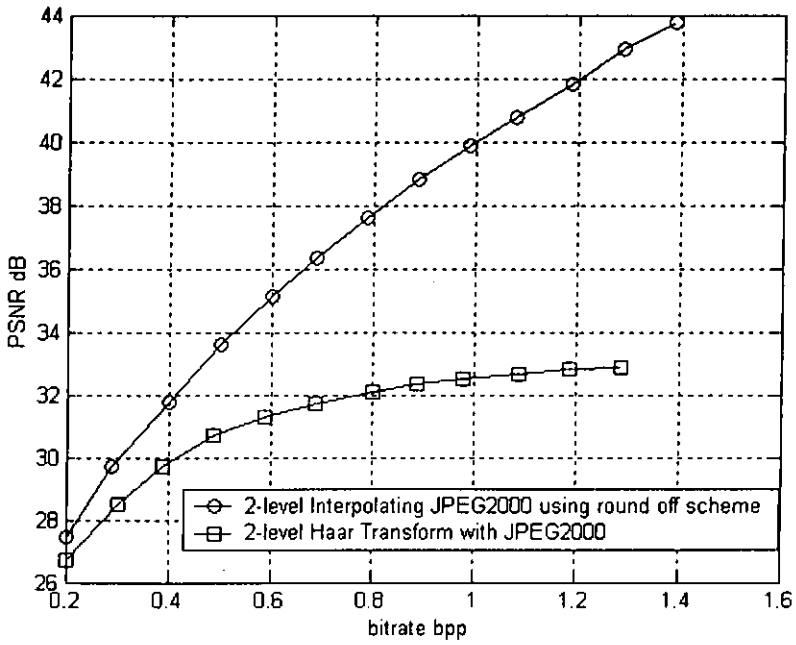


(c)

Fig. 5.6. A plot on PSNR against bit rate with different decomposition level at full frame rate, (a) *Carphone*, (b) *Foreman* and (c) *Claire*.

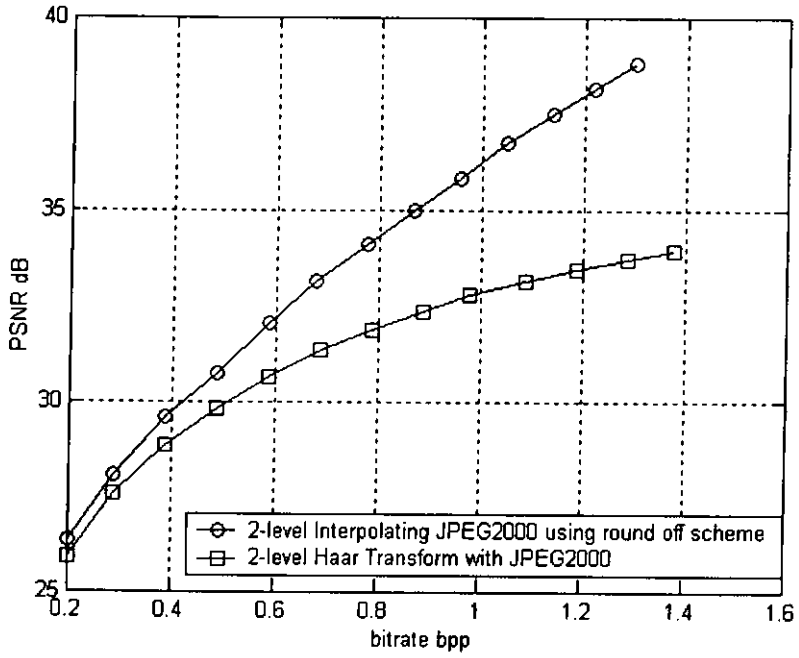


(a)

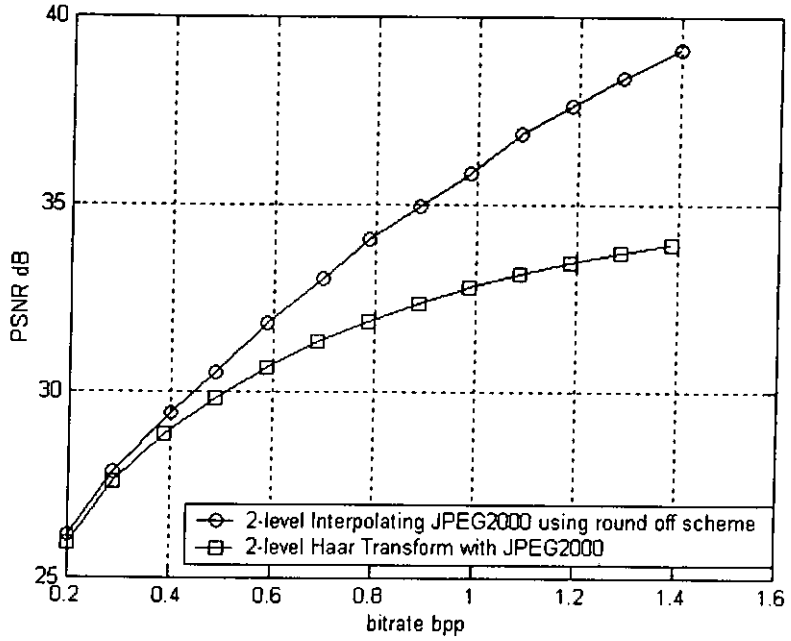


(b)

Fig. 5.7. Lower frame rate performance of *Carphone*, (a) 1 / 2 frame rate and (b) 1 / 4 frame rate.

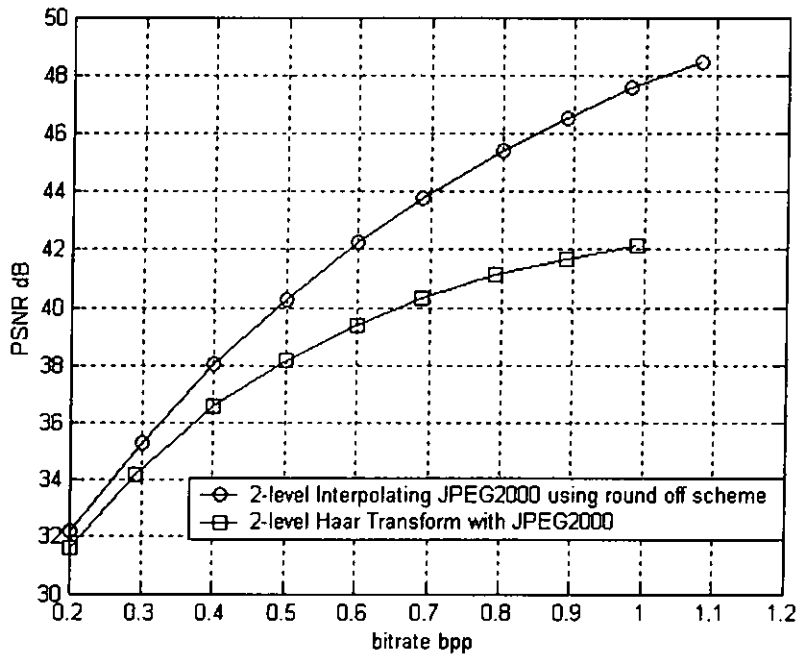


(a)

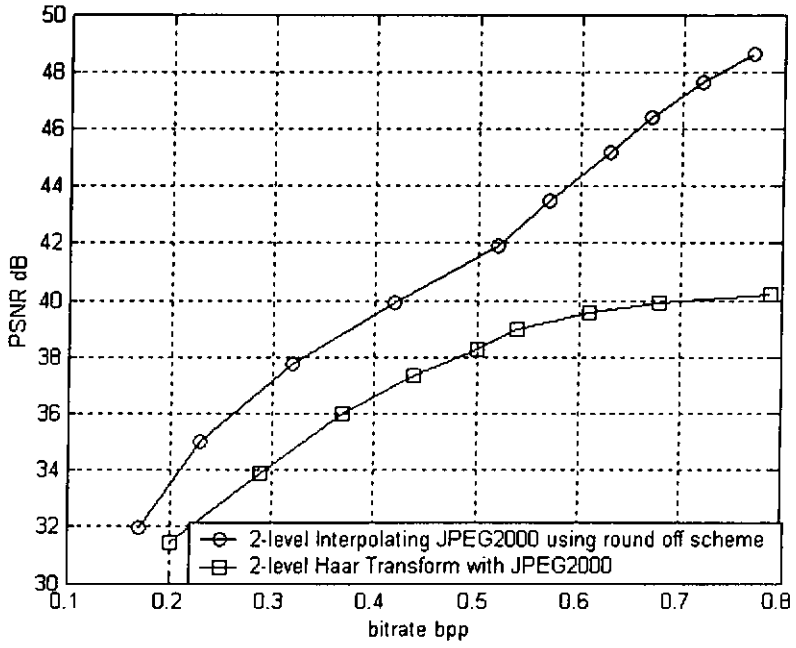


(b)

Fig. 5.8. Lower frame rate performance of *Foreman*, (a) 1 / 2 frame rate and (b) 1 / 4 frame rate.



(a)



(b)

Fig. 5.9. Lower frame rate performance of *Claire*, (a) 1 / 2 frame rate and (b) 1 / 4 frame rate.

Figures 5.7 – 5.9 further illustrate that the proposed approach substantially outperforms the traditional TSB in providing lower frame rate videos. It is foreseeable since the lower frame rate videos given by the traditional TSB is only a moving average of video frames while the ones given by the proposed approach are exactly the original video

frames at lower frame rates. Figure 5.10 gives a snap-shot of the lower frame rate videos given by both approaches. The improvement of the proposed approach is predictable.

(a) (b)

Fig. 5.10 a and b. Frame 57 of the decoded *Carphone* using the proposed approach (a) and the traditional TSB approach with Haar wavelet basis (b). Temporal decomposition level $N = 2$.

Figures 5.11 and 5.12 further illustrate the improvement given by the proposed reversible rounding method. Results for two video sequences, *Carphone* and *Foreman*, are given. To deal with the floating point coefficients generated by the interpolating wavelet transform, both the scale up and reversible rounding methods are used and compared. It can be seen that by using the reversible rounding method, a consistent improvement in PSNR up to 1 dB is obtained as compared to just simply scaling up the coefficients. Figures 5.11 and 5.12 also show the results compared with Motion-JPEG2000, of which nothing is done to reduce the temporal redundancy. For these two video sequences with higher motion activity, the proposed approach can still effectively remove temporal redundancy and PSNR at 0.6bpp is improved by nearly 1.5dB for both sequences compared to the original Motion-JPEG2000. The proposed approach also compares with the H.263 video codec, where temporal redundancy is reduced by using the traditional motion compensated prediction technique. In the comparison, the H.263 codec is operated in consecutive INTRA and INTER frames mode. At a bit rate of less than 0.6bpp, the

motion compensated prediction plays an important role in removing temporal redundancy for the videos that have high motion activity, such as *Foreman* and *Carphone*. However, if the bit rate is further increased, the interpolating method is good enough to achieve better compression efficiency over using computationally intensive motion compensated prediction.

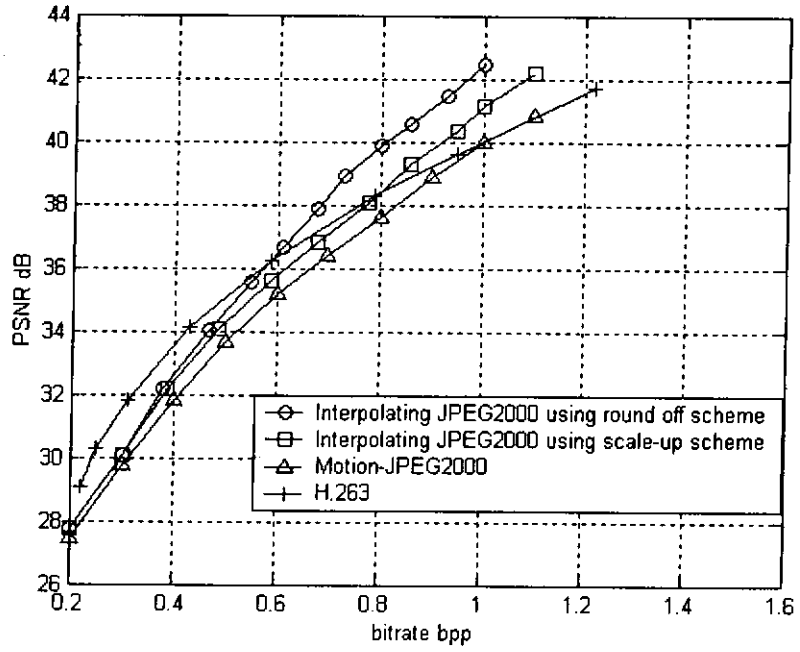


Fig. 5.11. A plot of PSNR(dB) against bit rate for *Carphone*. Decomposition level N=1. Full frame rate.

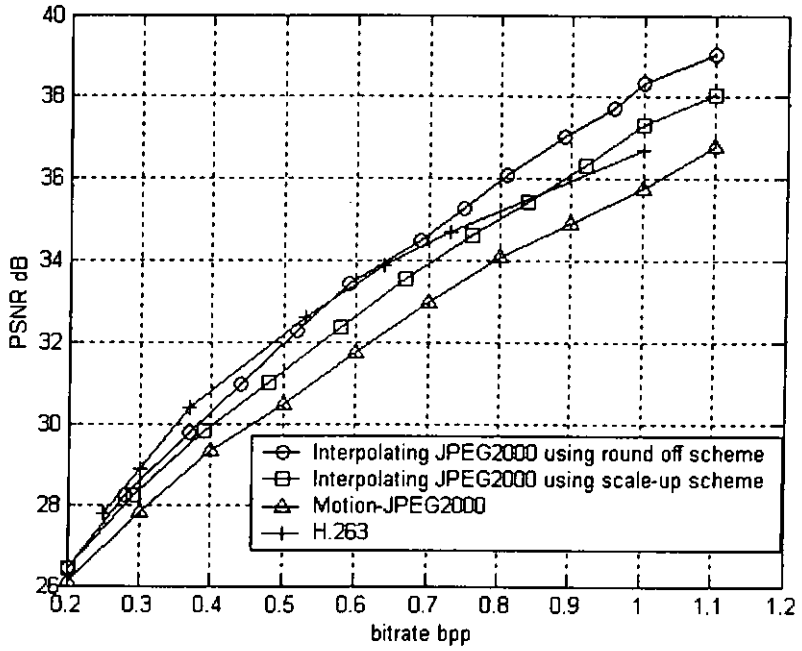


Fig. 5.12. A plot of PSNR(dB) against bit rate for *Foreman*. Decomposition level N=1. Full frame rate.

With the reversible rounding method, the proposed video codec can be further extended to lossless compression. Table 5.1 lists the bit rate for different testing video sequences compressed in lossless mode using the Motion-JPEG2000 codec and the proposed codec with scale-up and reversible round methods. The results show that the proposed method has a better performance in overall compression rate compared to Motion-JPEG2000. Again, the use of the reversible rounding method achieves a higher compression rate than just scaling up the coefficients to integers.

	Bit rate (bpp)		
Video sequence	Motion-JPEG2000	Interpolating JPEG2000 using scale up method	Interpolating JPEG2000 using reversible rounding off method
Carphone	3.9	4.19	3.62
Foreman	4.69	4.66	4.19
Claire	2.83	2.69	2.35

Table 5.1. The bit rate for lossless compression of video sequences using different compression methods.

5.6 Summary

In this chapter, we present a temporally scalable video coding system that is suitable for lossy and lossless video coding. With the proposed approach, the input video frames are first applied to an interpolating wavelet transform which generates video frames with reduced temporal redundancy in its high pass branch and lower rate video frames in its low pass branch. We further propose the reversible rounding method to convert the floating point transform coefficients into integers without loss of resolution. Experimental results show that the bit rate of the proposed approach is lower than the Motion-JPEG2000 codec in lossless compression. The difference in bit rate can be more than 10%. As for lossy compression, the PSNR of the reconstructed video frame at lower frame rates is substantially better than the traditional TSB approach which uses the Haar basis. Apart from compression efficiency, the proposed video coding system imposes no restriction on the image codec used for spatial coding. It means that it can benefit from the latest developments in image coding to further enhance the performance of the video coding system. This is the reason why, for the proposed system, all the advanced features of JPEG2000 can still be applied.

Chapter 6

Conclusions

6.1 General conclusions

In this work, the design and implementation of two important techniques in video coding, namely, content-based technique and temporal-scalable technique for low bit rate video coding are investigated. In this section, some conclusions are drawn in these aspects.

Firstly, we discuss the major techniques used in video compression in Chapter 2. We have reviewed some of the basic video compression techniques that have been adopted in current video coding standards. As for the second generation video compression, two important functionalities, content-based coding and temporal scalability are reviewed. We point out that problems still exist for the implementation of these two functionalities. For content-based coding, we are still facing a tradeoff between accurate representation of object shape and the required bandwidth. On the other hand, although various temporally scalable video coding algorithms have been proposed in recent years, inappropriate balance among of the degree of scalability, computational complexity and video quality are often found in these algorithms. These problems motivate us to develop better algorithms in the following chapters.

In Chapter 3, we present a content-based scalable H.263 video coding system that is suitable for low bit rate video applications. It is shown that under low bit rate condition, the proposed system can provide a consistent improvement in terms of PSNR when comparing with the conventional H.263 codec. Besides, the encoded video sequence can be decoded with conventional H.263 decoder which is different from other proprietary codec that requires a tailor-made decoder. The proposed system can be applied to low bit rate video applications, such as video conferencing systems and video surveillance systems with fixed camera setting.

While the proposed system allocates different amount of bits to the coding of foreground moving objects and static background, it is more desirable if we can further identify the speed of the moving objects and allocate different bit budgets to their coding. To achieve this, we need a more sophisticated feature extraction technique to classify the speed of the objects in real-time. This kind of technique is often application oriented. Hence in Chapter 4, we further apply the content-based H.263 video coding scheme to road traffic monitoring. Rather than only differentiating moving objects (i.e. cars in this application) from static background, we improve the coding scheme by further differentiating fast moving objects from slow moving objects and assigning different resource for their coding. In that approach, the location of lanes in a road traffic video is first determined by a new lane finding procedure. Two new techniques are then proposed for the classification of object speed. They are developed based on the observation that fast moving objects on the roads often have regular motion. Psychovisual thresholding technique is further applied to reduce the spatial redundancy when coding objects of different speed. As a result, it is shown that the bit rate of the proposed system decreases

as compared with the conventional H.263 codec. The decrease in bit rate can be more than 20%. Besides, the encoded video sequence can be decoded with conventional H.263 decoder, which is different from other proprietary codec that requires a tailor-made decoder. Besides road traffic monitoring, the proposed system can also be applied to other video surveillance systems with fixed camera setting.

As the final part of our study, the investigation on the temporal-scalable techniques for video coding is presented in Chapter 5. Traditional approaches implement temporal scalability by either introducing extra reference frames to the motion compensated prediction (MCP) video coding algorithms or simply switching to the temporal subband (TSB) video coding approaches. While the MCP approaches introduce extra complexity to the already complicated motion compensation process, the TSB approach may give a substantially degraded performance particularly for lower frame rate videos. In Chapter 5, we present a temporally scalable video coding system that is suitable for lossy and lossless video coding. With the proposed approach, the input video frames are first applied to an interpolating wavelet transform which generates video frames with reduced temporal redundancy in its high pass branch and lower rate video frames in its low pass branch. We further propose the reversible rounding method to convert the floating point transform coefficients into integers without loss of resolution. Experimental results show that the bit rate of the proposed approach is lower than the Motion-JPEG2000 codec in lossless compression. The difference in bit rate can be more than 10%. As for lossy compression, the PSNR of the reconstructed video frame at lower frame rates is substantially better than the traditional TSB approach which uses the Haar basis. Apart from compression efficiency, the proposed video coding system imposes no restriction on the image codec used for spatial coding. It means that it can benefit from the latest developments in image

coding to further enhance the performance of the video coding system. This is the reason why, for the proposed system, all the advanced features of JPEG2000 can still be applied.

Content-based coding and temporal-scalable techniques are two important techniques for compression efficiency and scalability. While content-based coding can provide high compression rate and retain reasonably good visual quality, the temporal scalability techniques can allow adaptation of video quality to different constraints incurred in the video transmission or retrieval processes. In this thesis, we have presented several new approaches to improve the performance when implementing these two techniques. Notwithstanding the current study is an academic one, certain results of this work have been applied to practical systems. For example, the content-based video coding system has been used in a demonstration of the new CDPD mobile system developed by an international mobile equipment provider in Hong Kong. This illustrates the academic and practical values of this study.

6.2 Future extensions

The results obtained in this work are just a small contribution to the development of video compression. While our work provides practical solutions to the problems mentioned in the previous chapters, it also incurs some possible research areas for further investigation. In this section, we discuss some of the possible extension to this work.

Pao Yue-Kong Library
PolyU • Hong Kong

6.2.1 Content-based video coding

For content-based coding, compression mainly comes from the proper bit allocation for video objects. Once we have the information about the special features of the video objects, video object plane (VOP) can be generated. The next important task will then be the coding of such VOPs. To achieve high compression efficiency, we require compression technique that is able to effectively code the various object features, such as shape, texture, and motion, etc. However, it is not a trivial task to have a compression technique that is suitable for the coding of all features. Indeed, we require the encoder to have the capability of encoding each feature using a multitude of compression techniques and choose one that is the most efficient. However, existing video codecs fail to carry out such operation. It will be fruitful for developing such dynamic coding system to achieve more powerful video data compression.

6.2.2 Temporal scalability

In recent years, multiwavelet is one of the hot topics in signal processing. Multiwavelet are wavelet bases with multiple scaling and wavelet functions. Unlike scalar wavelets, which have one scaling and wavelet functions, multiwavelets can simultaneously possess orthogonality and symmetry. They in general have compact support and higher number of vanishing moments. For these reasons, it is recently suggested to apply multiwavelets to the next generation image coding system. Nevertheless, as the input signal for multiwavelets requires 1-D vector input, any 2-D input signal such as image signal, must be transformed into two or more 1-D signals before taking the transform. Video signal, however, can be treated as a vector of images,

that fits naturally to the framework of multiwavelets. While the traditional subband video coding scheme is often criticized due to the low capability in reducing temporal redundancy, multiwavelets, which is capable of considering an area of pixels at a time, is expected to have a higher power in reducing temporal redundancy than the scalar wavelets in the subband video coding systems. Besides, the temporal scalability that is achieved by the current subband coding scheme is also possible in the multiwavelets system since they also have the multiresolution feature. Due to the nice properties of multiwavelets, we believe it is a promising direction for the development of new video codecs with powerful compression efficiency and high degree of temporal scalability by using multiwavelets.

BIBLIOGRAPHY

[1] T. Ebrahimi and M. Kunt, "Visual Data Compression for Multimedia Applications," IEEE Proceedings, vol. 86, no. 6, pp. 1109 –1125, June 1998.

[2] ISO/IEC JTX1 CD, 11172, (MPEG), "Information technology – Coding of moving pictures and associated audio for digital storage media up to about 1.5Mbit/s – Part 2" Coding of moving pictures information," 1991.

[3] "Information technology – Generic coding of moving pictures and associated audio information – Part 2: Video," ISO/IEC DIS 13818-2, MPEG-2, 1994.

[4] ITU Telecom., "Recommendation H.261 – Video codec for audiovisual services at p x 64 Kbit/s," Mar. 1993.

[5] ITU Telecom., Standardization Sector of ITU, "Video coding for low bitrate communication," ITU-T Recommendation H.263, Mar. 1996.

[6] ITU Telecom., Standardization Sector of ITU, "Video coding for low bitrate communication," Draft ITU-T Recommendation H.263 Version 2, Sept. 1997.

[7] G. Cote, B. Erol, , M. Gallant and F. Kossentini, "H.263+: video coding at low bit rates," IEEE Transactions on Circuits and Systems for Video Technology, vol. 8 no. 7 , pp. 849 – 866, Nov. 1998.

[8] Kai-Hong Ho and Daniel P. K. Lun, "Content-Based Scalable H.263 Video Coding Scheme for Road Traffic Monitoring," Proceedings, International Workshop on Multimedia Data Storage, Retrieval, Integration and Applications (MMWS'2000), pp. 163 – 170, Jan. 2000, Hong Kong.

[9] Kai-Hong Ho and Daniel P. K. Lun, "Content-based Scalable H.263 Video Coding for Road Traffic Monitoring," Submitted to IEEE Transactions on Multimedia.

[10] ISO/IEC JTC1/SC29/WG1N1696, "Motion-JPEG2000 Requirements and Profiles ver 2.0," March 2000.

[11] ISO/IEC JTC1/SC29/WG1N1803, "JPEG2000 Requirements and Profiles ver 6.3", July 2000.

[12] ISO/IEC JTC1 CD 1496-2 (MPEG-4), "Information technology – Coding of audio-visual object: Visual," Oct. 1997.

[13] E. Dubois, "The Sampling and Reconstruction of Time-varying Imagery with Application in Video Systems," IEEE Proceedings, vol. 73, no. 4, pp. 502 – 522, Apr. 1985.

[14] L. Torres and M. Kunt, Video Coding, The Second Generation Approach. Boston: Kluwer Academic, 1996.

[15] T. Sikora, "MPEG digital video-coding standards," IEEE Signal Processing Magazine, vol. 14, no. 5, pp. 82 – 100, Sept. 1997.

[16] W. H. Chen, C. H. Smith and S. C. Fralick, "A Fast Computational Algorithm for the Discrete Cosine Transform," IEEE Transactions on Communications, vol. COM-5, pp. 1004 – 1009, Sept. 1977.

[17] B. G. Lee, "A New Algorithm to Compute the Discrete Cosine Transform," IEEE Transactions on Acoustics, Speech and Signal Processing, vol. ASSP-32, no. 6, pp.1243 – 1245, Dec. 1984.

[18] E. Feig and S. Winograd, "Fast Algorithms for the Discrete Cosine Transform," IEEE Transactions on Signal Processing, vol. 40, no. 9, pp. 2174 – 2193, Sept. 1992.

[19] D. Nister and C. Christopoulos, " An Embedded DCT-Based Still Image Coding Algorithm," IEEE Signal Processing Letters, vol. 5, no. 6, pp. 135 – 137, June 1998.

[20] A. Said and W. A. Pearlman, "A New, Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees," IEEE Transactions on Circuits and Systems for Video Technology, vol. 6, no. 3, pp. 243 – 250, June 1996.

[21] Z. Xiong, O. Guleryuz and M. T. Orchard, "A DCT-based Embedded Image Coder," IEEE Signal Processing Letters, vol. 3, no. 11, pp. 289 – 290, Nov. 1996.

[22] S. G. Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 11, no. 7, pp. 674 – 693, July 1989.

[23] I. Daubechies, "Orthonormal bases of compactly supported wavelets," Comm. Pure Appl. Math., vol. 41, pp. 909 – 996, Nov. 1998.

[24] L. M. Po and W. C. Ma, "A Novel Four-step Search Algorithm For Fast Block Motion Estimation," IEEE Transactions on Circuits and Systems for Video Technology, vol. 6, no. 3, pp. 313 – 317, June 1996.

[25] R. Li, B. Zeng and M. Liou, "A New Three-step Search Algorithm for Block Motion Estimation," IEEE Transactions on Circuits and Systems for Video Technology, vol. 4, no. 4, pp. 438 – 442, Aug. 1994.

[26] T. Koga and K. Iinuma, "Motion-compensated Inter-frame Coding for Video Conferencing," in Proc. NTC81, New Orleans, LA, pp. C9.6.1 – 9.6.5, Nov. 1981.

[27] S. Kappagantula and K. R. Rao, "Motion Compensated Inter-frame Image Prediction," IEEE Transactions on Communications, vol. COM-33, pp. 1011 – 1015, Sept. 1985.

[28] J. R. Jain and A. K. Jain, "Displacement Measurement and its Application in Inter-frame Image Coding," IEEE Transactions on Communications, vol. COM-29, pp. 1799 – 1808, Dec. 1981.

[29] M. Ghanbari, "The Cross-search Algorithm for Motion Estimation," IEEE Transactions on Communications, vol. 38, no. 7, pp.950 – 953, July 1990.

[30] I. H. Witten, R. M. Neal and J. G Cleary, "Arithmetic Coding for Data Compression," Comm. of the ACM, vol. 30, no. 6, pp. 520 – 540, Jun. 1987.

[31] D. Minoli, "Video Dialtone Technology: Digital Video over ADSL, HFC, FTTC and ATM," McGraw-Hill, New Your, USA, 1985.

[32] B. J. Kim and W. A. Pearlman, "An Embedded Wavelet Video Coder using Three-dimensional Set Partitioning in Hierarchical Trees (SPIHT)," Proceedings, Data Compression Conference (DCC'97), pp. 251 – 260, 1997.

[33] T. Meier and K. N. Ngan, "Video Segmentation for Content-based Coding," IEEE Transactions on Circuit and Systems for Video Technology, vol. 9, no. 8, pp. 1190 – 1203, Dec. 1999.

[34] R. Mech and M. Wollborn, "A Noise Robust Method for Segmentation of Moving Objects in Video Sequences," IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-97), vol. 4, pp. 2657 – 2660, 1997.

[35] R. Talluri, K. Oehler, T. Bannon, J. D. Courtney, A. Das and J. Liao, "A Robust, Scalable Objected-based Video Compression Technique for Very Low Bit Rate Video Coding," IEEE Transactions on Circuits and Systems for Video Technology, vol. 7, no. 1, pp. 221 – 233, Feb. 1997.

[36] S. Colonnese and G. Russo, "User interaction modes in semi-automatic segmentation: Development of a flexible graphic user interface in Java," ISO/IEC JTC1/SC29/WG11 MPEG98/m3320, Tokyo, Japan, Mar. 1998.

[37] J. G. Choi, M. Kim, J. Kwak, M. H. Lee and C. Ahn, "User-assisted video object segmentation by multiple object tracking," ISO/IEC/JTC1/SC29/WG11 MPEG98/m3349, Tokyo, Japan, Mar. 1998.

[38] P. Gerken, "Object-based Analysis-synthesis Coding of Image Sequences at Very Low Bit Rates," IEEE Transactions on Circuits and Systems for Video Technology, vol. 4, no. 3, pp. 228 – 235, June 1994.

[39] H. Katata, N. Ito and H. Kusao, "Temporal-scalable Coding Based on Image Content," IEEE Transactions on Circuits and Systems for Video Technology, vol. 7, no. 1, pp. 52 – 59, Feb 1997.

[40] T. Fukuhara, K. Asai and T. Murakami, "Very Low Bit-rate Video Coding with Block Partitioning and Adaptive Selection of Two Time-differential Frame Memories," IEEE Transactions on Circuits and Systems for Video Technology, vol. 7, no. 1, pp. 212 – 220, Feb 1997.

[41] B. K. P. Horn and B. G. Schunck, "Determining Optical Flow," Artificial Intelligence, vol. 17, pp. 185 – 203, 1981.

[42] F. Meyer and S. Beucher, "Morphological segmentation," J. Visual Commum. Image Representation, vol. 1, pp. 21 – 46, Sept. 1990.

[43] P. Salembier, P. Brigger, J. R. Casas and M. Pardàs, "Morphological Operators for Image and Video Compression," IEEE Transactions on Image Processing, vol. 5, no. 6, pp. 881 – 898, June 1996.

[44] G. Lilienfield and J. W. Woods, "Scalable High-definition Video Coding," Proceedings, International Conference on Image Processing, vol. 2, pp. 567-570, 1995.

[45] D. Taubman and A. Zakhor, "Multirate 3-D Subband Coding of Video," IEEE Transactions on Image Processing, vol. 3, no. 5, pp. 572 – 588, Sept. 1994.

[46] D. L. Donoho, "Interpolating wavelet transform," Dept. Statistics, Standford University, Tech. Rep., Oct. 1992.

[47] G. Deslauriers and S. Dubuc, "Symmetric iterative interpolation processes," Constructive Approximation, 5, pp. 49 – 68, 1989.

[48] W. Sweldens, "The Lifting Scheme: A Custom-design Construction of Biorthogonal Wavelets," Appl. Comput. Harmon. Anal., vol. 3, no. 2, pp. 186 – 200, 1996.

[49] J. R. Ohm, "Three-dimensional Subband Coding with Motion Compensation," IEEE Transactions on Image Processing, vol. 3, no. 5, pp. 559 – 571, Sept. 1994.

[50] K. H. Ho and P. K. Lun, "On Temporally Scalable Video Coding Using Interpolating Wavelet Transform," Submitted to IEEE Transactions on Circuits and Systems for Video Technology, 2001.

[51] R. G. Gallager, "Variations on a Theme by Huffman," IEEE Transactions on Information Theory, vol. IT-24, pp. 668 – 674, Nov. 1978.

[52] K. M. Rose and A. Heiman, "Enhancement of One-dimensional Variable Length DPCM Images Corrupted by Transmission Errors," IEEE Transactions on Communications, vol. 37, no. 4, pp. 373 – 379, Apr. 1989.

[53] D. Marpe, G. Blattermann, G. Heising and T. Wiegand, "Video Compression Using Context-based Adaptive Arithmetic Coding," Proceedings, International Conference on Image Processing, vol. 3, pp. 558 – 561, 2001.

[54] E. Baum, V. Harr and J. Speidel, "Improvement of H.263 Encoding by Adaptive Arithmetic Coding," IEEE Transactions on Circuits and Systems for Video Technology, vol. 10, no. 5, pp. 797 – 800, Aug. 2000.

[55] Y. Linde, A. Buzo and R. M. Gray, "An Algorithm for Vector Quantizer Design," IEEE Transactions on Communications, vol. 28, pp. 84 – 95, Jan. 1980.

[56] L. Torres and E. Arias, "Stochastic vector quantization of images," Proceedings, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP-92), vol. 3, pp. 385 – 388, 1992.

[57] H. Freeman, "On the encoding of arbitrary geometric configuration," IRE Trans. Electron. Comput., vol. EC-10, pp. 260 – 268, June 1961.

[58] H. Samet, "Region representation: Quadtrees from Boundary Codes," Commun. ACM, vol. 23, no. 3, pp. 163 – 170, Mar. 1980.

[59] M. Hőtter and R. Thoma, "Image segmentation based on object oriented mapping parameter estimation," Signal processing, vol.15, no.3, pp. 315-334, Oct. 1988.

[60] R. Talluri, K. Oehler, T. Bannon, J. D. Courtney, A. Das and J. Liao, "A Robust, Scalable Objected-based Video Compression Technique for Very Low Bit Rate Video Coding,", IEEE Transactions on Circuits and Systems for Video Technology, vol. 7, no. 1, pp. 221 – 233, Feb. 1997.

[61] Software tmn3.2 at ftp://dspftp.ece.ubc.ca/pub/tmn/ver-3.2.

[62] W. Y. Ma and B. S. Manjunath, "Edge flow: A framework of boundary detection and image segmentation," Proceedings, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 744 – 749, 1997.

[63] C. A. Christopoulous, W. Philips, A. N. Skodras and J. Cornelis, "Segmented image coding: Techniques and experimental results," Signal Processing: Image Communication 11, pp. 63 – 80, 1997.

[64] F. Dufaux and F. Moscheni, "Segmentation based motion estimation for second generation video coding techniques," in: L. Torres, M. Kunts (Eds), "Video Coding," Kluwer Academic Publishers, Boston, pp. 219 – 263, 1996.

[65] Skufstad Kurt and Jain Ramesh, "Illumination independent change detection for real world sequence," Computer Vision, Graphics and Image Processing, 46 pp. 387 – 299, 1989.

[66] Kai-Hong Ho and Daniel P. K. Lun, "Content-Based Scalable H.263 Video Coding Scheme for Road Traffic Monitoring," Proceedings, International Workshop on Multimedia Data Storage, Retrieval, Integration and Applications (MMWS'2000), pp. 163 – 70, Jan. 2000, Hong Kong.

[67] J. Soh, B. T. Chun and M. Wang, "Analysis of Road Image Sequences for Vehicle Counting," Proceedings, IEEE International conference on Intelligent Systems for the 21st Century, vol. 1, pp. 679 – 683, 1995.

[68] B. D. Stewart, I. Reading, M. S. Thomson, T. D. Binnie, K. W. Dickinson and C. L. Wan, "Adaptive Lane Finding in Road Traffic Image Analysis," Proceedings, IEE Conference on Road Traffic Monitoring and Control, pp. 133 – 136, 1994.

[69] Y. Ninomiya, Y. Ohtsuka, Y. Izumi, S. Gohshi and Y. Iwadate, "Present Status of MUSE," Proceedings, 2nd International Workshop on Signal Processing of HDTV, pp. 579 – 602, 1988.

[70] M. G. Perkins and T. Lookabaugh, "A Psychophysically Justified Bit Allocation Algorithm for Subband Image Coding Systems," Proceedings, IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-89), vol. 3, pp. 1815 – 1818, 1989.

[71] J. T. Kim, H. J. Lee and J. S. Choi, "Subband Coding Using Human Visual Characteristics for Image Signals," IEEE Journal on selected area in communications, vol. 11, no. 1, pp. 59 – 64, 1993.

[72] A. N. Netravali and B. G. Haskell, "Digital pictures: representation, compression and standard," Second Edition, Plenum Press, New York, 1995.

[73] K. N. Ngan, K. S. Leong, and H. Singh, " Adaptive Cosine Transform Coding of Images in Perceptual Domain," IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 37, no. 11, pp. 553 – 559, 1989.

[74] D. L. McLaren and D. T. Nguyen, "Removal of Subjective Redundancy from DCT-coded Images," IEE Proceedings - Communications, Speech and Vision, vol. 138, no. 5, pp. 345 – 350, Oct. 1991.

[75] K. Uz, M. Vetterli and D. LeGall, "Interpolative Multiresolution Coding of Advanced Television with Compatible Subchannels," IEEE Transactions on Circuits and Systems for Video Technology, vol. 1, no. 1, pp. 86 – 99, March 1991.

[76] C. Podilchuk, N. Jayant, and N. Farvardin, "Three Dimensional Subband Coding of Video," IEEE Transactions on Image Processing, vol. 4, no. 2, pp. 125 – 139, Feb. 1995.

[77] Y. Chen and W. A. Pearlman, 'Three-dimensional subband coding of video using the zero-tree method," Proceedings, SPIE Visual Communications and Image Processing, vol. 2727, pt. 3, pp. 1302 – 1312, 1996.

[78] G. J. Conklin and S. S. Hemami, "A Comparison of Temporal Scalability Techniques," IEEE Transactions on Circuits and Systems for Video Technology, vol. 9, no. 6, pp. 909 – 919, Sept. 1999.

[79] S. J. Choi and J. W. Woods, "Three-dimensional subband/wavelet coding of video with motion compensation," Proceedings, SPIE Visual Communications and Image Processing, vol. 3024, pt. 1, pp. 96 – 104, 1997.

[80] P. L. Shui and Z. Bao, "Recursive Biorthogonal Interpolating Wavelets and Signal-Adapted Interpolating Filter Banks," IEEE Transactions on Signal Processing, vol. 48, no. 9, pp. 2585 – 2593, Sept. 2000.

[81] Z. Shi, G. W. Wei, D. J. Kouri, D. K. Hoffmand and Z. Bao, "Lagrange Wavelets for Signal Processing," IEEE Transactions on Image Processing, vol. 10, no. 10, pp. 1488 – 1508, Oct. 2001.