LOW COMPLEXITY VIDEO CODING

DESIGN AND ANALYSIS OF LOW COMPLEXITY VIDEO CODING FOR REALTIME COMMUNICATIONS

BY

INSU PARK, B.Sc, M.Sc, M.A.Sc

A THESIS

SUBMITTED TO THE SCHOOL OF COMPUTATIONAL ENGINEERING & SCIENCE

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

DOCTOR OF PHILOSOPHY (2010) McMaster University

(Computational Engineering & Science) Hamilton, Ontario, Canada

TITLE: Design and Analysis of Low Complexity Video Coding for Realtime Communications

AUTHOR: Insu Park

M.A.Sc, (Electrical Engineering) McMaster University

M.Sc, (Electronic Engineering) Sogang University

B.Sc, (Control and Measurement Engineering) Chosun University

SUPERVISOR: Dr. David Capson

NUMBER OF PAGES: xix, 167

To both my parents and
my lovely wife Younhyoung

Abstract

Video coding standards have been designed to support many applications such as broadcasting systems, movie industries and media storage. All video coding standards try to reduce data in video sequences as much as possible by exploiting spatial and temporal redundancies. Although those video coding standards are suitable for a wide variety of applications, some applications require low encoder complexity specifically for real time video encoding. Most of the computational complexity of a video encoder can be attributed to the motion estimation function.

Motion estimation using multiple reference frames is widely used as the basis for recent video coding standards (eg. H.264/AVC) to achieve increased coding efficiency. However, this increases the complexity of the encoding process. In this thesis, new techniques for efficient motion estimation are proposed. A combination of multiple reference frame selection and image residue-based mode selection is used to improve motion estimation time. By dynamic selection of an initial reference frame in advance, the number of reference frames to be considered is reduced. In addition, from examination of the residue between the current block and reconstructed blocks in preceding frames, variable block size mode decisions are made. Modified initial motion vector estimation and early stop condition detection are also adopted to speed up the motion estimation procedure. Experimental results compare the performance

of the proposed algorithm with state of the art motion estimation algorithms and demonstrate significantly reduced motion estimation time while maintaining PSNR performance.

In addition a new side information generation algorithm using dynamic motion estimation and post processing is proposed for improved distributed video coding. Multiple reference frames are employed for motion estimation at the side information frame generation block of the decoder. After motion estimation and compensation, post processing is applied to improve the hole and overlapped areas on the reconstructed side information frame. Both median filtering and residual-based block selecting algorithms are used to deal with hole and overlapped areas, respectively. The proposed side information method contributes to improving the quality of reconstructed frames at the distributed video decoder. The average encoding time of the distributed video coding is shown to be around 15% of H.264 inter coding and 40% of H.264 intra coding. The proposed side generation algorithm is implemented in a frequency domain distributed system and tested throughout various test sequences. The proposed side information based distributed video coding demonstrates improved performance compared with that of H.264 intra coding.

Experimental implementations of the proposed algorithms are demonstrated using a set of video test sequences that are widely used and freely available.

Acknowledgements

I would like to sincerely thank and acknowledge my supervisor, Dr. David Capson, for his generous support, excellent technical and personal guidance, and for providing me with an opportunity to pursue my research. Particular thanks also go to my committee members Dr. Allan D. Spence and Dr. Suzanna Becker for their valuable comments, advice and feedback. Special thanks are also due to my friends and colleagues for their kind help and friendship. I thank Laura Kobayashi for her help.

Lastly, but not least, I am especially grateful for the continuous support of my family. Their endless support, encouragement and love have always been the source of achievement in my higher education.

Glossary of Terms

ASIC	Application Specific Integrated Circuits
bps	bit per second
CABAC	Context Based Adaptive Binary Arithmetic Coding
CAVLC	Context Adaptive Variable Length Coding
CIF	Common Intermediate Format
DCT	Discrete Cosine Transform
DVD	Digital Versatile Disc
DSC	Distributed Source Coding
DVC	Distributed Video Coding
ET	Encoding Time
FPGA	Field Programmable Gate Array
fps	frames per second
GOP	Group of Pictures
HDTV	High Definition Television
HVS	Human Visual System
IDCT	Inverse DCT
ISO	International Standard Organization
ITU	International Telecommunication Union
ITU-T	International Telecommunication Union Telecommunication Sector
JPEG	Joint Picture Experts Group
kbps	kilo bits per second
LDPC	Low Density Parity Check
MAE	Mean Absolute Error
MB	Macroblock
MC	Motion Compensation
ME	Motion Estimation

MET	Motion Estimation Time
MPEG	Moving Picture Expert Group
MSE	Mean Squared Error
MV	Motion Vector
PSNR	Peak Signal to Noise Ratio
QCIF	Quarter CIF
QP	Quantization Parameter
RD	Rate Distortion
RDO	Rate Distortion Optimization
SAD	Sum of Absolute Difference
SATD	Sum of Absolute Transformed Difference
VCD	Video Compact Disc
VCEG	Video Coding Expert Group
VHDL	VHSIC Hardware Description Language
VHSIC	Very High Speed Integrated Circuit
VLC	Variable Length Coding
VLSI	Very Large Scale Integration
YCbCr	1 component for luminance(Y), 2 components for chrominance (CbCr)

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The purpose of this chapter is to provide the motivation for the current research as well as brief descriptions of target applications. This chapter includes an overview of previous work related to the current research, a summary of the contributions of the current work, and the outline of this thesis.

1.1 Motivation

Real time video encoding has been a challenging research topic in the last few decades because many new applications have emerged in our lives. Since the early 1970's several video coding standards have been proposed to encode video sequences. International organizations such as the *Moving Picture Expert Group* (MPEG) and *Video Coding Expert Group* (VCEG) were established to develop standards for video coding as well as image coding. The video coding standards from MPEG and VCEG are found in real life applications such as *Digital Video Disc* (DVD) adopting MPEG-2 and *Video Compact Disc* (VCD) using MPEG-1.

Conventional video coding standards have been designed to support many applications such as broadcasting systems, movie industries and media storage. All of these video coding standards try to reduce data redundancy existing in video sequences as much as possible with available techniques. Although these video coding standards are suitable for various applications, some applications require low encoder complexity for real time video encoding. Video communication via mobile phone and surveillance camera are examples of new applications requiring real time video encoding. To achieve real time video encoding, the conventional video coding standards now require encoder resources that are too high.

High encoder complexity is a critical problem for encoders with low hardware resources. For example, it is difficult to encode more than two *frames per second* (fps) when QCIF (176×144) format video sequences are encoded with a high level processor and ample memory. When CIF (352×288) format video sequences are used, the average encoding rate is less than one fps. Considering commercial applications which require from 24 fps to 30 fps, the encoding rate of the conventional video standard is far away from the required encoding rate.

To speed up the encoder, designing a low complexity video encoder is necessary. Thus designing a low complexity video encoder has been a major topic in the design of video encoders area recently. Since most encoder complexities are from *motion estimation* (ME) and *motion compensation* (MC), optimizing the ME and MC procedures is widely adopted to implement low complexity video encoders. Reducing the complexity of the video encoder can be achieved through many algorithms at the expense of coding performance.

The purpose of this research is to design and evaluate the performance new methods to reduce encoder complexity while keeping both the quality of reconstructed video sequences and the size of encoded video data similar to the latest video coding standard (H.264/AVC). In the following subsection, examples of target applications of a low video encoder are described.

1.2 Previous Work

In this section, previous research into a low complexity video encoder are introduced. To begin, various ME algorithms, which have been proposed to reduce the encoder complexity, are introduced briefly. Further details about efficient ME algorithms are discussed in Chapter 4. Secondly, a new video coding paradigm, known as *distributed video coding* (DVC), is briefly introduced. The theoretical background and implementation issues of DVC are described in Chapter 5.

1.2.1 Efficient Motion Estimation

Efficient ME algorithms for video coding have been proposed in the literature for a long period. A common approach is to designing search patterns to reduce the number of search points in a search window. Using designed search pattern, the computation of ME is effectively reduced by calculating the cost function of the specific points on each search pattern instead of all points in the search window. Although it provides the best performance, a full search is the most computationally heavy algorithm.

Various search patterns such as the cross search [4], four step search [5], three step search [6, 7, 8], gradient descent search [9], rood pattern search [10, 11] and diamond

search [12, 13] are examples from the recent literature. Recently the hexagon search algorithm was proposed in [14]. Four most popular search patterns are depicted in Figure 3.3. Also, combinations of more than one search pattern were introduced to reduce ME time [15]. In H.264/AVC video encoding both the 3D hexagon search [16] and fast integer pel hexagon search which are called UMHexagonS [17] were tested and provided good performance. All proposed search pattern algorithms check only some specific points on the search pattern instead of all points in a search window.

In video coding, reference frames are the previous or future frames, which can be used in ME for the current frame. Figure 3.4 shows the previous reference frames $(t-1, t-2, \cdots t-n)$ for the current frame (t). Multiple reference frames based ME generally provides better performance than single frame based ME. The encoder complexity of multiple reference frames based ME is linearly increased in proportion to the number of reference frames. When ME is performed through multiple reference frames, efficient reference selection algorithms contribute to reduce the ME time. Since the H.264/AVC video encoder supports multiple reference frames based ME, most reference frame selection algorithms were tested on H.264/AVC reference software. Some approaches of fast reference selection algorithms can be found in [18, 19, 20, 21, 22, 23] and [24].

While the video coding standards such as H.261, H.263, MPEG-1, 2, and 4 support *macroblock* (MB) size based ME, H.264/AVC introduced variable MB size based ME to improve the coding efficiency. Variable MB mode is described in Figure 3.2. In variable MB mode, ten different MB modes exist for each MB. Supporting the variable MB mode also increases the complexity of the encoder because it compares the cost function of each mode and chooses one MB mode, which has the smallest

value of the cost function. Thus efficient MB mode selection algorithms have been introduced by skipping the calculation of cost functions in some MB modes instead of checking all possible MB modes. Fast mode decision algorithms for H.264/AVC can be found in [25] and [26]. Wu *et al.* [27] defined spatial homogeneity areas using edge information and stationary regions with a MB difference. Then homogeneity information is used for mode decision in block and sub-block mode.

Spinsante *et al.* [28] defined thresholds for the early termination during mode decision. In Spinsante's work, threshold T_1 is the average rate distortion of all the MBs encoded with skip mode. Threshold T_2 is based on the neighboring MBs encoded with the same mode. Then thresholds were compared with neighboring blocks. The performance can be improved by updating the threshold with integer values. In Gao and Lu's algorithm [29], the threshold for skip T_{skip} is updated whenever there is a new skip mode. They also compared the cost function of the block mode with the cost function of the sub-block mode to decide whether to make it sub-block mode or not. The early predicted zero motion block detection algorithm is another factor in reducing ME time. Both early skip and selective intra mode decisions are introduced to reduce the complexity by Choi *et al.* [30]. Other efficient mode decision algorithms for video encoding can be found in [31, 32] and [33].

Fast ME can be achieved by reducing the search window size. Liu [34] proposed a dynamic search range decision based ME algorithm for video coding. It is shown that dynamic search range decision based ME reduces ME time by 47% compared to full search based ME. An adaptive search window size algorithm was proposed by Goel *et al.* [35] and tested on the H.264/AVC video encoder. Goel's algorithm eliminates almost 90% to 95% of candidate search points. Another search window

size decision algorithm for H.264/AVC video encoder was proposed by Bailo *et al.* [36]. By adopting search window size decision algorithm they reduced about 50% to 60% of encoding time.

1.2.2 Distributed Video Coding

Although the algorithms introduced in the previous section contribute to reduce encoder complexity, there is still high computational complexity in the video encoder. Most encoder complexity is from the ME procedure. *Distributed video coding* (DVC) is a new video coding paradigm, which is totally different from the concept of conventional video coding standards. DVC was proposed by Girod *et al.* in [2]. The main difference between DVC and conventional video coding standards is that there is no process to remove temporal redundancy at the encoder of DVC. Thus the DVC encoder does not perform ME. Usually the performance of DVC is worse than the performance of conventional video coding standards. But DVC's encoder complexity is much less than the complexity of the conventional video encoders.

Due to its significantly lower encoder complexity the concept of DVC was adapted to many applications. Puri *et al.* [37] designed a wireless sensor network using DVC. Guillemot *et al.* [38] built a mono-view and multi-view video coding system based on a distributed coding algorithm. An example of video compression and error resilience of DVC can be found in [39]. In [40] Valera and Velastin proposed a smart distributed video surveillance system.

As we mentioned earlier the performance of DVC is worse than the performance of conventional video coding standards with ME. Thus many algorithms have been proposed to improve the coding performance of DVC. The DVC system is depicted in

Table 1.1: Video standards and their major features

Name	Completion time	Major features
H.261	1990	For video conferencing, data rate is 64 kbps~1.9 Mbps
MPEG-1	1991	For CD-ROM applications, data rate is up to 1.5 Mbps
MPEG-2	1994	For Digital TV, data rate is 2~ 5 Mbps
H.263	1995	For very low bit rate applications, data rate is below 64 kbps
MPEG-4	1999	For multimedia and interactive video applications
H.264/AVC	2003	For a wide range of video applications

Figure 5.1. In DVC system, side information is the information used in decoder when reconstruct the Wyner-Ziv frame. The outputs of side information generation procedure are influence on both the Wyner-Ziv frame reconstruction and the soft input estimation module of turbo decoder. Well estimated side information frame request fewer bit from the encoder buffer to generate the current Wyner-Ziv frame.

Therefore, the quality of side information significantly influences the RD performance of DVC system. Improving side information at the decoder side is the most popular way to improve the quality of reconstructed video frames at the decoder [41]. Interpolation and extrapolation are widely used side information generation algorithms. Usually side information based on interpolation is of better quality than side information based on extrapolation [42]. Li and Delp [43] employed the ME concept at the decoder to generate side information and provided good quality reconstructed video frames.

More details about the background of DVC systems and implementation issues are discussed in Chapter 5.

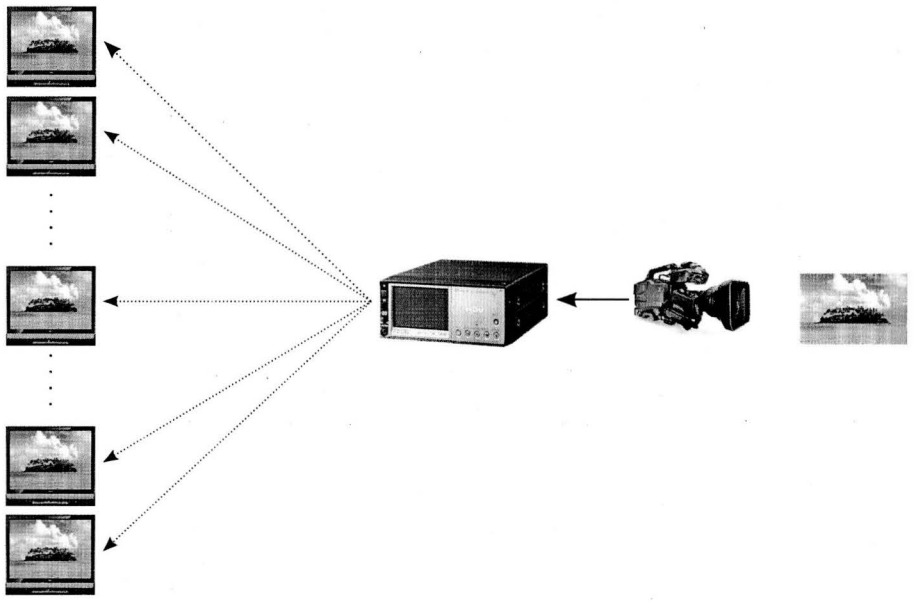


Figure 1.1: The example of one time encoding and many times decoding application.

1.3 Application Examples

1.3.1 The Application of the Current Video Coding Standards

As mentioned earlier the target application of the conventional video coding standards is a "one time encoding and many time decoding" system. So when the standards were designed real time encoding was out of consideration. Table 1.1 summarizes the conventional video coding standards and their major features including target applications. Figure 1.1 shows an example of an application of conventional video coding standards. A video scene is captured by a video camera and saved on the server to perform several processes such as frame interpolation, gamma correction, de-interlacing, etc. Then the video data is delivered to the decoder side or to customers.

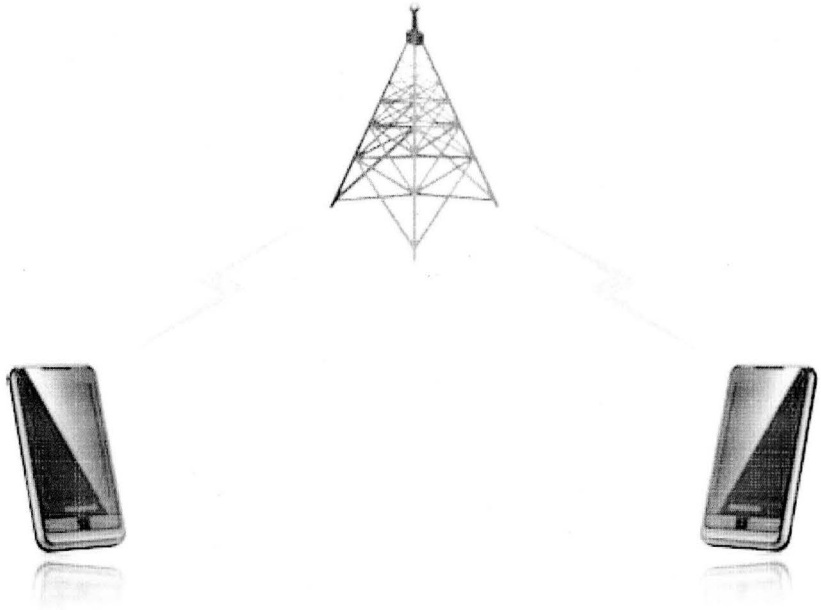


Figure 1.2: The example of video telephony on mobile devices.

At the decoder, the compressed video data is decompressed and played as many times as the customer wants to play. In those applications, real time decoding is more important than real time encoding. Thus most complexities exist on the encoder side to satisfy the requirement of the conventional video coding applications.

1.3.2 Video Telephony on Mobile Devices

In mobile devices such as cell-phones and PDA based smartphones, battery life is one of the most important factors in the consumer's consideration. When those mobile devices perform video encoding and decoding as well as audio encoding and decoding, the battery life is related to the complexity of video codec, audio codec,

size of panel (LCD), etc. Figure 1.2 depicts an example of communication through a mobile phone. The hardware resources of the latest mobile phone are around 800 MHz for the processor and a few giga bytes of memory. While those mobile phones provide almost 500 hours of stand-by time, the battery life typically is limited to five hours when audio encoding and decoding procedures are performed.

Video encoder and decoder pairs require more computational complexity than audio encoder and decoder pairs. Thus the battery life can be worse when video communication is performed on those devices. Figure 1.2 can be implemented using DVC with conventional video coding systems such as H.263 and H.264/AVC for video telephony. In those scenarios, two end users require only distributed video encoders and a H.26x decoder. During video communication, at first, the video stream is compressed using a DVC system encoder. The coded video bitstream is sent to the server having a DVC to H.26x transcoder. Since the server has high performance processing unit and large memory resources, the transcoding can be performed in real time. Then the server sends coded H.26x video bitstreams to the destination mobile device and the H.26x decoder decompresses the coded bitstream. Thus the mobile device achieves low complexity video encoding and decoding.

1.3.3 Multi View with Camera Array

In some applications, an object is recoded using multi arrayed cameras. After encoding an object or scene the camera sends compressed video data to a central station. In this case, there is high correlation between the video sequences acquired from neighboring cameras. Thus at the central decoder, joint decoding can be implemented.

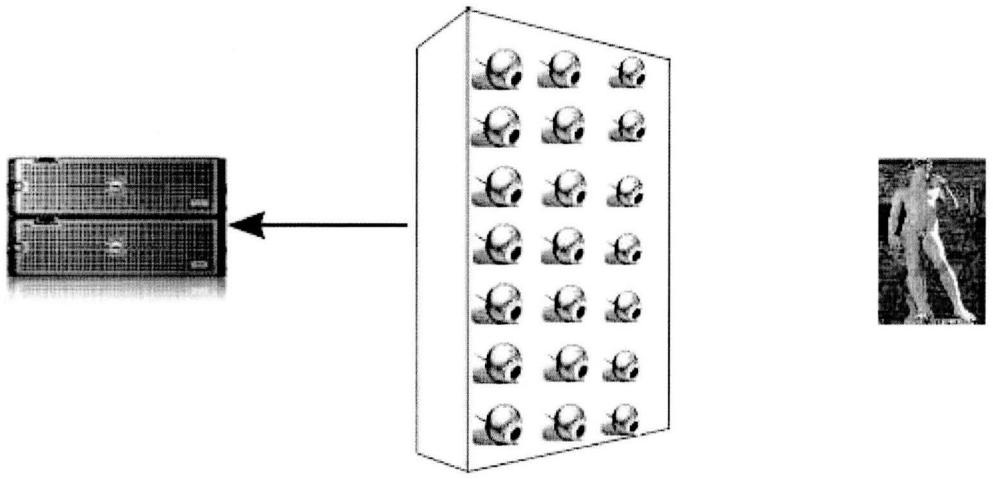


Figure 1.3: The example of multi view with camera array.

Also the processing and memory resources of a camera are limited. For this application, DVC is useful to encode the video frame at each camera to reduce the total complexity of the camera array.

Figure 1.3 shows an example multi view camera array. In Figure 1.3 the array of multiple cameras is used to transmit the video stream of a specific scene to the destination such as a file server and central decoder. Since the process unit and memory resources in each camera are restricted, a conventional video encoder, which requires high computational complexity, is not suitable. Also due to the high correlation among video streams from neighboring cameras, DVC is more preferable than the conventional video coding scheme.

1.4 Research Overview

This work explores the design of a low complexity video encoder for real time applications. To achieve this goal, an efficient ME algorithm for the H.264/AVC video encoder is proposed. Also an improved DVC system is proposed to achieve the goal. The following sections describe the proposed methods to improve video encoder complexity followed by the list of contributions and the outline of the thesis.

1.4.1 Proposed Methods for Low Power Video Encoder

In the proposed method, for the low power video encoder, different approaches are adopted to reduce encoder complexity by the dynamic reference selection algorithm, residue based mode decision algorithm and early termination algorithm. The proposed algorithms are implemented on the H.264/AVC reference software designed by the joint video team of MPEG and VCEG. By using the information from neighboring blocks, which is defined as the surrounding block of the current block, the encoder chooses the initial reference frame. The MV search procedure is extended to another previous and future frame if the cost function keeps decreasing.

Residue is the difference between the current block and the block in the reference frame. In H.264/AVC variable block mode is supported to improve the coding efficiency. In the proposed algorithm, the residue information of the current block is used to find the best mode of the current block before calculating the cost function of all possible block modes. At first residue information is used to decide whether the current block is suitable for skip mode or not. When a block is coded with skip mode, the encoder does not need to send any information such as the MV and the entropy code of DCT coefficients. Also the residue information is used to decide the sub-block

mode. Early termination during variable block mode based ME is another method to reduce the ME complexity. For early termination, threshold(s) for an acceptable error should be defined in advance. In the proposed algorithm, the upper bound and lower bound of the threshold are decided using the residue information of neighboring blocks.

In this work, DVC is introduced for the low complexity video encoder as well. Compared to conventional video coding standards such as MPEG-x and H.26X, DVC is a new coding paradigm of encoders with lower complexity but higher decoder complexity. While conventional video coding standards perform inter encoding and intra decoding, the DVC performs intra encoding and inter decoding structures. Thus the complexity of DVC is much lower than the intra and inter modes of the conventional video coding standards. DVC provides solutions for applications requiring extremely low encoder complexity and/or low power consumption. Instead of its low encoder complexity, DVC has low coding performance compared to the inter mode of the conventional video coding performance.

A new decoding algorithm of DVC is proposed to improve the coding standard. Both multiple reference frames based ME and post processing are adopted to improve the quality of side information at the decoder. After ME processing at the decoder, the linear interpolation method is used to estimate the MV of the current block. In post processing, a neighboring block containing minimum distortion is used to fill out the hole area and overlapped area.

1.4.2 List of Contributions

There are two major contributions for designing low complexity video encoders in this work. The first is efficient motion estimation achieved by initial reference frame selection, mode decision and early stop algorithms. The second is designing an improved side information frame for the DVC system. The contributions of this work are listed below.

- An efficient motion estimation algorithm for H.264/AVC adopting multiple reference frames is proposed. The proposed algorithm investigates neighboring macroblocks to estimate the optimal reference frame, motion vector and thresholds for a stop condition. Simulation results show that the proposed motion estimation outperforms the conventional motion estimation based on full search ME algorithms. Also, the PSNR of the proposed motion estimation is shown to be comparable with that of UMHexagonS based motion estimation while requiring less motion estimation time. The proposed motion estimation has better performance when the video sequence has large motion. Thus, the proposed algorithm is appropriate for applications in video compression such as sport sequences. This work has been done and shown in [44].

- The residue between a current MB and reconstructed previous MB was investigated for efficient mode decision in H.264/AVC. The information from residue is useful for mode decision in variable block size motion estimation. Compared with calculating a cost function (J), obtaining residue requires substantially less computation. The proposed algorithm was verified by testing several standard video sequences with various characteristics. Experimental results show

the proposed algorithm reduces motion estimation time by up to 25%. Further improvement can be achieved at the expense of increasing bit rate. The simulation results of this work were published in [45].

- A fast ME algorithm for multiple reference frame coding in H.264/AVC was proposed. To reduce the complexity of ME, an initial reference selection algorithm, an estimation of the initial MV and early stop condition were considered together. Also, the difference between the current macroblock and reconstructed previous frames was used for fast mode decision. From the simulation results, the proposed ME algorithm reduces motion estimation time significantly while the deterioration of PSNR can be considered to be minimal. The proposed ME algorithm has better performance for video sequences with small motion and static or slowing varying backgrounds. Since a fewer number of reference frames are required, and it uses variable block size, the proposed encoder reduces the complexity for ME. This situation is enhanced when there is high correlation between current and neighboring macroblocks.

- Because DVC employs independent frame coding and does not remove the temporal redundancy at the encoder, the coding efficiency of DVC is known to be worse than the conventional video coding standards with inter coding. To improve the coding efficiency, the DVC investigates temporal redundancy at the decoder side and provides the information to the Wyner-Ziv frame decoder block. Side information consists of the number of required bits as well as the approximated pixel values and plays a key role. Thus improving the quality of the side information frame is a critical issue in the DVC system to achieve high quality reconstructed frames. A new side information generation algorithm

was proposed to improve the coding efficiency of DVC. Motion estimation using multiple reference frames approximates the current Wyner-Ziv frame when there is large motion of the object and a complicated background in a frame. Also, simple mean filtering was used to recover hole areas on the side information frame. Employing a residue based overlapped block selection algorithm is another character of the proposed side information frame algorithm. The encoding complexity of DVC is reduced by more than 30%. The performance of DVC based on the proposed side information algorithm provides an acceptable quality of reconstructed frames compared to conventional video coding standards.

1.4.3 Outline of the Thesis

This work describes a low computational complexity video encoder for real time applications. Throughout the work the proposed methods are implemented using the reference software of the H.264/AVC. The coding performance of the proposed methods is compared to the performance of H.264/AVC in terms of *rate distortion* (RD). The encoder complexity is measured by encoding time and ME time. Throughout the experiments, standard test sequences provided by MPEG and VCEG are used and the test sequences are listed in Appendix A.

This first chapter is intended as an introduction to the research motivation, the major applications of the research and previous work. In this chapter, two methods, which are efficient ME and the DVC system, are briefly described. One time encoding and many time decoding applications are introduced as examples of the conventional video coding system. Video telephony on mobile devices and multi view

camera arrays are described as examples of low complexity video encoder systems. Also our contribution to low complexity video encoders and the list of publications are mentioned in this chapter.

Chapter 2 is devoted to describing the conventional video coding standards, which are proposed by international MPEG and VCEG. Major tools for video coding such as ME, DCT, quantization and entropy coding are introduced first. Then video coding standards and their specific features are described in detail. The encoder structure and tools used in H.261, H.263, MPEG-1, MPEG-2, MPEG-4 and H.264/AVC are also discussed in this chapter. For comparison of the performance of video coding standards, the concepts of rate and distortion are introduced. Then the performance video coding standards are compared and discussed.

The computational complexity issue of the conventional video coding standards is discussed in Chapter 3. In the inter mode of the conventional video coding standard, most complexity is from the ME and MC. Depending on the mode decision and the number of reference frames, ME and MC take around 50% to 80% of encoder complexity. The complexity of each function of the encoder is investigated using H.264/AVC reference software. The results show why an efficient ME algorithm is important to design a low complexity video encoder.

In Chapter 4, efficient ME methods are introduced as part of the low complexity video encoder. Some famous and widely used search pattern algorithms are described. Then ME using information from neighboring blocks is introduced. Initial reference frame and initial MV are estimated from the neighboring block information. Also thresholds for early stop condition are defined using the characteristics of the neighboring blocks. In H.264/AVC supporting variable macroblock mode causes

the encoder complexity to increase. The approach to reduce encoder complexity by a mode decision algorithm is discussed. First, the analysis of mode decision of the H.264/AVC reference software is investigated. Then the residue image between the current block and the reference block is defined and used for a smart mode decision algorithm inter mode decision. Simulation results of the proposed algorithms and discussion are provided at the end of the chapter.

Chapter 5 introduces a new video coding paradigm called distributed video coding. Two well known theories, which are the Slepian-Wolf and Wyner-Ziv theories, are introduced to explain the theoretical background of the DVC system. Then both pixel domain and transform domain Wyner-Ziv frame coding are discussed and compared in the following sections. Side information is an important process in a DVC system to achieve high performance. The ME based side information generation algorithm as well as the extrapolation and interpolation based side information generation algorithm are discussed.

Then the proposed improved side information generation algorithms are introduced. Both multiple reference frames based ME and post processing are adopted to improve the side information frame. The proposed algorithm is plugged into the conventional DVC system and the results are discussed at the end of the chapter.

Chapter 6 provides the summary of this with colclusions. This chapter ends with a discussion of hardware implementation issues and possible directions for future work.

Chapter 2

Video Coding Standards

Video standards have been proposed to make international standards to satisfy the requirement of consumer products. Each video coding standard has target applications and major features to achieve the required coding performance. In this chapter various video coding standards and tools for video coding are introduced. Also comparisons of the performance of different video coding standards are provided.

2.1 Introduction

Multimedia applications are a part of our everyday life. In multimedia applications, video contents are more popular than audio contents, and have the largest amount of data among the different multimedia applications. The huge amount of video data is the main obstruction for video data storage as well as video data communication. Thus reducing the amount of video data is required for real applications. A lot of studies have been done to reduce the data of video applications. International standard organizations such as MPEG and ITU-T were established to make video coding

standards.

Most video coding techniques are from image coding algorithms. Video coding is the extension of image coding into the time domain. While images have only spatial redundancy, video data has temporal redundancy as well as spatial redundancy. Thus video coding tries to remove both spatial and temporal redundancy while image coding reduces spatial redundancy. To remove spatial redundancy, video coding uses different techniques from image coding: image coding uses the wavelet transform, whereas video coding uses DCT.

Since the early 1980's video coding experts have proposed efficient video coding algorithms. Each video coding standard, with its own application area, has been proposed as the requirements of its application changed. Currently there are two series of video coding standards: H.26x, which were proposed by ITU-T, and MPEG-x, which were proposed by MPEG. The purpose of the ITU-T video coding standard is different from the purpose of the MPEG video coding standard. While the ITU-T video coding standard places more importance on the functionality of communication, the MPEG video coding standard emphasizes the function of storage and broadcasting.

In this chapter, the existing video coding standards such as MPEG-1,2,4, H.261, H.263, and H.264/AVC will be introduced. Then the performance of different coding standards will be compared to each other. For objective comparison, performance measurement tools are presented. Since most video codings are lossy, the distortion of reconstructed video should be considered as well as the size of compressed video data. *Peak signal to noise ratio* (PSNR) is used for the distortion measure. The mathematical meaning of the PSNR also will be investigated in this chapter.

This chapter is organized as follows. Basic tools and principles are described

in Section 2.2. A series of video coding standards is introduced in Section 2.3. In Section 2.4, the performance measurement tools are explained, and different video coding standards are compared in terms of the performance measurement tools, and this chapter is finalized with a summary in Section 2.5.

2.2 Tools for Video Coding

In video sequences, there are both spatial and temporal redundancies. Temporal redundancy exists between consecutive frames, while spatial redundancy exists in a frame. Video coding standards remove both temporal and spatial redundancy to achieve a high compression ratio. ME and transformations are widely used to investigate temporal redundancy and spatial redundancy, respectively. A transformation such as DCT is applied to the residue image acquired using ME. After the transformation, the resultant coefficients are quantized using a quantization matrix or quantization scale factor. The quantized coefficients are then scanned in a predefined order. The scanned coefficients are coded with entropy coding. In the following section, each tool is described in detail.

2.2.1 Motion Estimation

All video codings perform block based coding. First, an input frame is divided into block of a predefined size. The basic unit of video coding is a *macroblock* (MB). ME is the process of finding the best matched area within the preceding block from a reference frame of the current block. Using ME, the current block can be represented with the displacement, which is called the *motion vector* (MV), and the difference

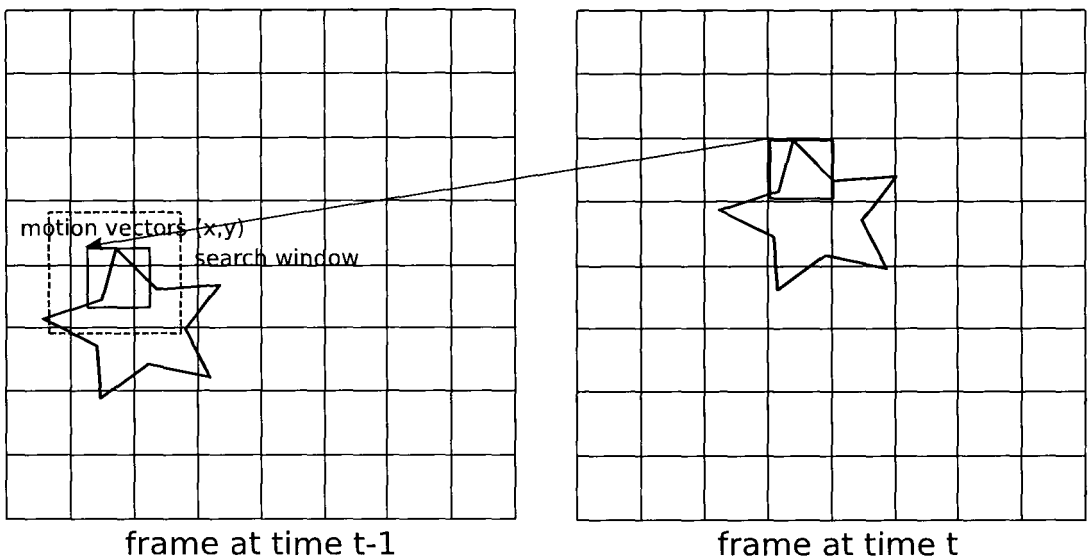


Figure 2.1: Motion estimation to find the best match of the current block from previous frame.

between the current block and the reference block, which is called the residue. Figure 2.1 depicts the concept of ME finding MV. To find the best matching area, a block distortion measure is used. The block distortion measure is defined as:

$$E_{(x,y)} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |f_t(i,j) - f_{t-1}(i+x, j+y)| \tag{2.1}$$

where $f_t(i,j)$ is the intensity value of the frame at the position (i,j), and x and y are MVs in the horizontal and vertical directions, respectively, and $N \times N$ is the block size.

Then the MVs (mv_x, mv_y) minimizing the block distortion $E_{(x,y)}$ can be represented as

$$\{mv_x, mv_y\} = \underset{x,y\in[-W,+W]}{\operatorname{argmin}} E_{(x,y)} \tag{2.2}$$

where W is a search window size. If all points in the search window are compared the ME algorithm is called full search based ME. Since full search based ME requires intensive computational complexity, efficient motion search algorithms have been proposed. The diamond search [12], and hexagon search [14] are examples of search patterns reducing the complexity of ME. Diamond search pattern and hexagon search pattern are depicted in Figures 3.3 (c) and (d), respectively.

2.2.2 Discrete Cosine Transform

A highly correlated image in the spatial domain can be de-correlated by transforming the data into the frequency domain. The *discrete cosine transform* (DCT) is a very popular transform in both image and video coding for this purpose. DCT de-correlates the image data very well compared to other transforms. After de-correlation, each coefficient of the transformed data is compressed independently. The forward DCT (FDCT) with length N is defined as

$$F(u) = \sum_{x=0}^{N-1} f(x)C(u)\cos\left[\frac{(2x+1)u\pi}{2N}\right] \tag{2.3}$$

for $u = 0, 1, 2, \cdots, N-1$. Then the *inverse discrete cosine transform* (IDCT) is given by

$$f(x) = \sum_{u=0}^{N-1} F(u)C(u)\cos\left[\frac{(2x+1)u\pi}{2N}\right] \tag{2.4}$$

for $x = 0, 1, 2, \cdots, N-1$. The coefficient $C(u)$ is

$$C(u) = \begin{cases} \sqrt{\frac{1}{N}} & \text{if} \quad u = 0 \\ \sqrt{\frac{2}{N}} & \text{otherwise} \end{cases} . \tag{2.5}$$

After normalizing, the FDCT coefficient $F(u)$ is the average value of sample data when $u = 0$. This coefficient is the DC coefficient; otherwise the coefficients are AC coefficients.

Since a video frame is a data set in *two dimensions* (2D) it is worth defining FDCT and IDCT pairs in 2D. The 2D FDCT is the direct extension of FDCT to two dimensions and is expressed as

$$F(u,v) = \sum_{y=0}^{N-1} \sum_{x=0}^{N-1} f(x,y) C(u) C(v) \cos\left[\frac{(2x+1)u\pi}{2N}\right] \cos\left[\frac{(2y+1)v\pi}{2N}\right] \tag{2.6}$$

for $u, v = 0, 1, 2, \cdots, N-1$. The 2-D IDCT is defined as

$$f(x,y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F(u,v) C(u) C(v) \cos\left[\frac{(2x+1)u\pi}{2N}\right] \cos\left[\frac{(2y+1)v\pi}{2N}\right] \tag{2.7}$$

for $x, y = 0, 1, 2, \cdots, N-1$. The coefficients $C(u)$ and $C(v)$ are the same as the coefficients of the 1D DCT which is given by Equation 2.5. In terms of implementation, a 2D FDCT can be performed with the 1D FDCT in the horizontal direction and then in the vertical direction. This property makes it easy to implement the 2D FDCT in real applications. The 2D FDCT has excellent de-correlation properties for video data, and compacts the information of highly correlated video data into low frequency areas. The energy compaction property is very useful for video coding when the transformation works with quantization.

2.2.3 Quantization

The DCT does not contribute to data compression itself. It is only when quantization is combined with DCT that the compression is achieved. The quantization procedure involves dividing DCT coefficients with the predefined values in quantization matrix. Different quantization values are used for different DCT coefficients, and the quantized DCT coefficients are truncated into integer values. The quantization step sizes are designed to emulate these of the human visual system. In the frequency domain, low frequencies represent the average brightness of the input frame while high frequencies are related to sharp edges in a frame. Since human vision is more sensitive to the low frequency components than the high frequency components, the quantization step sizes for low frequency components are smaller than the quantization step sizes for high frequency components.

2.2.4 Entropy Coding

After scanning, the quantized DCT coefficients are compressed using entropy coding schemes. Entropy coding is a lossless data compression technique. In entropy coding, the code length of a symbol is based on the probability of the symbol. Typically shorter code lengths are used for the symbols with high probability. Two of the most commonly used entropy coders are Huffman coding and arithmetic coding. In video coding standards usually run-length coding is used. Different code tables can be defined for different parameters.

Table 2.1: The resolution of different video sequence formats. The resolution of chrominance is calculated using the assumption that the sampling ratio of the video frame is 4:2:0

Format	Luminance		Chrominance	
	horizontal	vertical	horizontal	vertical
SQCIF	128	96	64	48
QCIF	176	144	88	72
SIF	352	240	176	120
CIF	352	288	176	144
4CIF	704	576	352	288
16CIF	1408	1152	704	576

2.3 Video Coding Standards

In this section, several video coding standards are reviewed. First, the concept of intra and inter modes, which only exist in video coding, are introduced to help further understand video coding standards. The major features of those standards are discussed in the following sections.

2.3.1 Intra and Inter Mode

Different video coding standards take different video sequence formats for the input source. The spatial resolution of different video formats are in Table 2.1. The resolution of video frames is directly related to the size of coded data. Each video frame has different types of coding mode based on the existence of ME and ME direction. If a frame is coded without ME, the frame is called intra frame (I-frame or I-picture). The I-frame is coded using only DCT, quantization, and entropy coding. If a frame

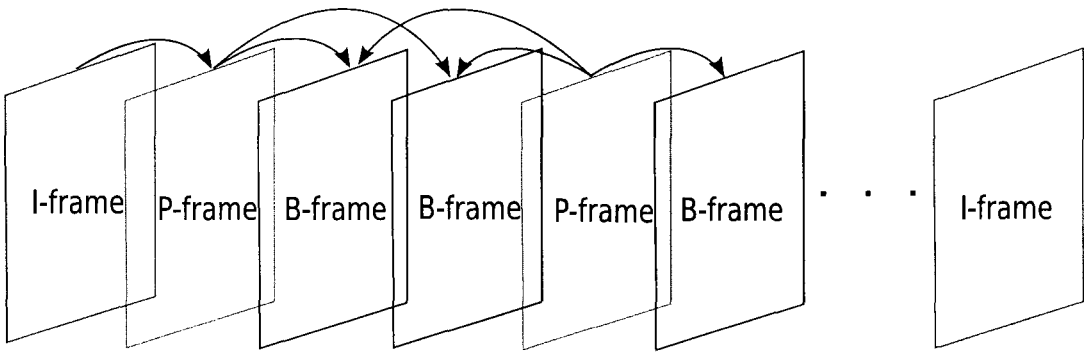


Figure 2.2: The different directions of the motion estimation of I-, P-, and B-frame coding.

is coded with ME and MC using only previous frames, the frame is called one directional inter frame (P-frame or P-picture). The bi-directional inter frame (B-frame or B-Picture) is a frame which is coded with ME and MC using both previous and future frames. In P-frame coding only the I-frame can be the reference frame, while both the I- and P-frame can be the reference frame in B-frame coding. Figure 2.2 shows the direction of ME for each frame mode.

Each video frame is divided into MBs which are the basic unit of video processing. The size of a MB is 16×16. All MBs in the I-frame are coded with intra mode, whereas MBs in P- and B-frame can be coded with either intra mode or inter mode. Each MB has four luminance blocks (Y) and two chrominance blocks Cb, Cr.The size of each block is 8×8. While Y component represents grey data, Cb and Cr components represent colour data of a pixel. Since colour data are not sensitive as much as grey data the colour data are down sampled to reduce data size. Thus there is a different sampling ratio among grey and colour data. The sampling ratio is expressed using three digits. For example, the sampling ratio is 4:4:4 if both grey and colour

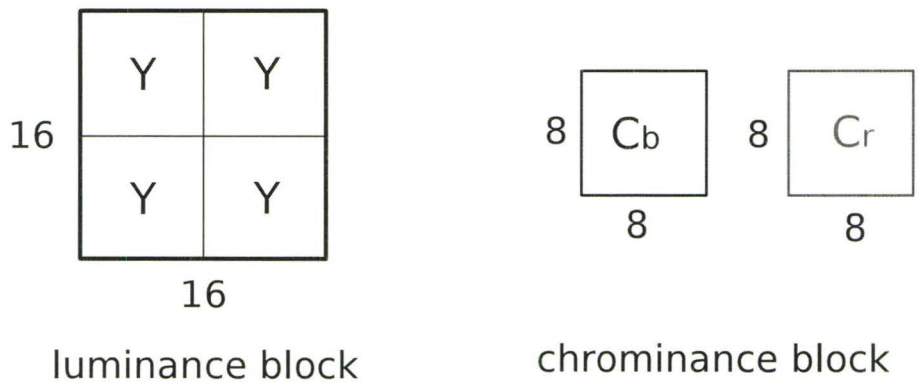


Figure 2.3: The structure of the macroblock when the video format has a 4:2:0 sampling ratio.

data are represented without down sampling. Figure 2.3 depicts the structure of a MB when the chrominance data of the video frame is sampled with 4:2:0 ratio.

2.3.2 H.261

The H.261 video coding standard was proposed by ITU-T in the early 1990's. The main application of H.261 is real time video conferencing and video telephony. The real time requirement restricts the complexity of the encoder. Thus the structure of H.261 is simpler than other video encoders. The H.261 supports only the CIF and QCIF video format. The range of target frame rate is from 7.5 to 30 *frames per second* (fps). The supported transmission rates are in multiples of 64 kilo bits per second (kbps). The target of H.261 is compressing a CIF format video sequence of 37 Mbps to 128 kbps and QCIF format sequence of 9 Mbps to 64 kbps. H.261 supports both I- and P-frame modes but not B-frame mode.

The H.261 encoder structure is the hybrid of ME/MC and 2D DCT coding. Each

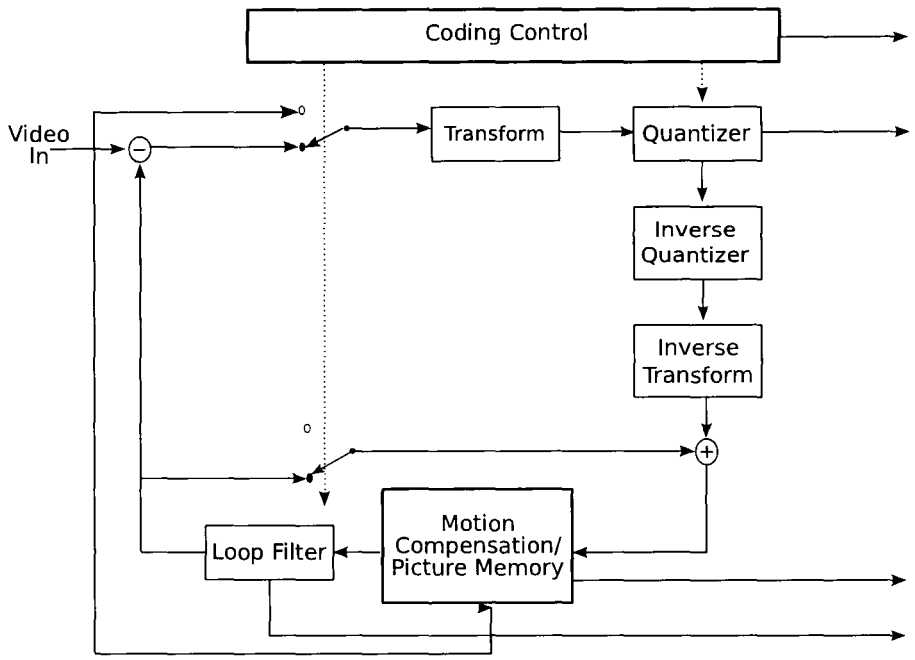


Figure 2.4: The structure of the H.261 encoder.

block, which has the size of 8×8, is transformed independently. After 2D DCT, quantization is applied to round the transform coefficients to integer values. For quantization, H.261 applies a quantization factor instead of a quantization matrix. Only one quantizer step size of 8 is defined for intra DC coefficients. For AC coefficients, 31 step sizes are defined.

In the H.261, the ME is based on a MB. The ME is performed on only luminance MBs. Thus, in P-frame coding, each MB has only one MV, which has integer accuracy. The MV of corresponding chrominance blocks is half of the MV of the luminance MB. The maximum search window size for MVs is ± 15 in horizontal and vertical directions. Only one reference frame is used for ME. To minimize the prediction error, H.261 adopted an optional loop filter. Variable length coders are defined

to code the quantized coefficients, the MVs, and control parameters. The structure of the H.261 encoder is depicted in Figure 2.4.

2.3.3 H.263

The H.263 video coding standard was approved in 1996. The H.263 is the next version of the H.261. Thus the basic algorithm of H.263 video is similar to that of H.261. Above all, the H.263 standard was designed for very low bit rate applications such as video telephony and video conferencing. The target bit rate is $p - \times$ 64 kbps. The H.263 supports five frame formats, which are SQCIF, QCIF, CIF, 4CIF and 16CIF. Both interlaced and progressive modes of video sequences can be used as an input source.

Besides the tools of the H.261 encoder, some advanced techniques are added to improve coding efficiency. Thus the coding efficiency of H.263 is better than H.261 but H.263 has more computational complexity. The main components of the H.263 encoder are transform, motion compensated prediction, quantization and variable length coding which are similar with H.261. Figure 2.5 depicts the encoder structure of H.263. The new features adopted in H.263 include: half-pixel precision, unrestricted MVs, syntax-based arithmetic coding, advanced prediction, and a PB-frames coding mode.

Half-pixel Precision

In the H.263 encoder, half-pixel precision motion compensation is used to improve the accuracy of the MV, while H.261 encoders use integer-pixel accuracy motion compensation. Using linear interpolation of the pixel value of integer-pixel points,

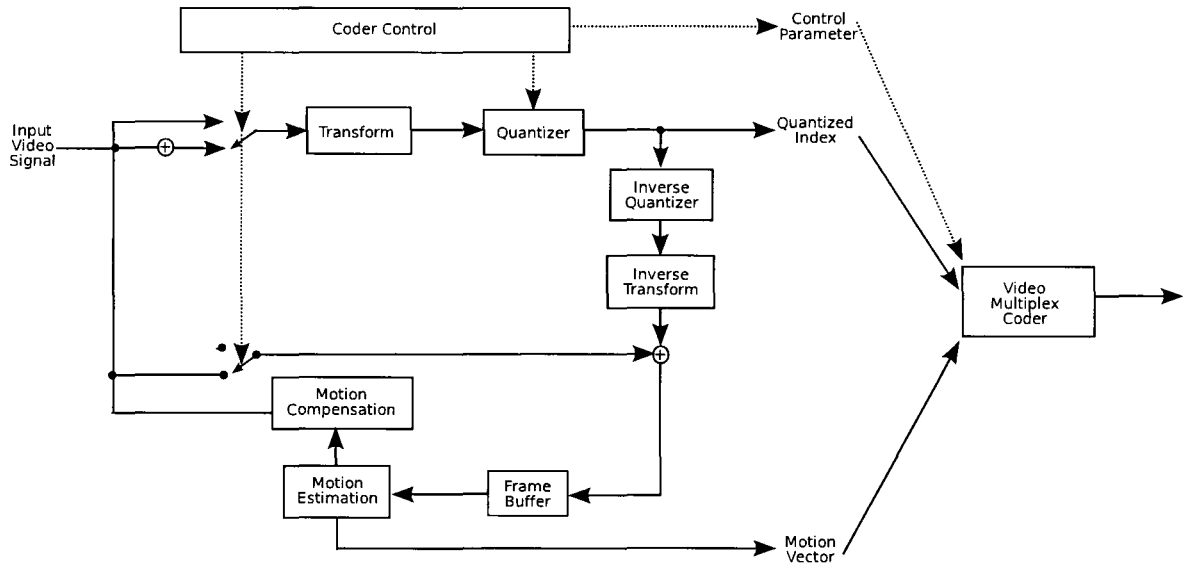


Figure 2.5: The structure of the H.263 video encoder.

the pixel value of half-pixel points is calculated. Half-pixel precision provides more precise MVs than integer pixel ME.

Unrestricted Motion Vectors

Unrestricted MVs allow ME to find MVs from outside of the reference frame. In the H.261 encoder, the MVs are restricted to the inside of the reference frame. The encoder assumes the pixel values outside of the reference frame as the boundary pixel value. The unrestricted MVs can lead to larger MVs. For example, MV ranges are extended to be [-32, 32] instead of [-16, 16].

Syntax-based Arithmetic Coding

Although arithmetic coding is optional, it is useful to reduce the bit rate in some video sequences. To achieve better efficiency, the H.263 encoder replaces the standard variable length coder with a syntax based coder. The coding tables are generated by training a large number of test sequences.

Advanced Prediction

In advanced prediction mode, the ME of the H.263 encoder finds a MV for each 8×8 block of inter MB. In advanced prediction mode, one or four MVs can be used to represent a MB. The number of MVs is based on the prediction error. Also overlapped motion compensation for a luminance block is supported.

PB-frames

A PB-frame consists of one P-frame and one B-frame. In a PB-frame, the P-frame is coded with ME/MC using a previous I- or P-frame, while the B-frame is coded with ME/MC using an I- or P-frame or P-frame in another PB-frame. Note that the H.261 does not support B-frame mode. In PB-frame mode, one MB has 12 blocks: six blocks are from the P-frame and six blocks are from the B-frame.

2.3.4 MPEG-1

The objective of the MPEG-1 standard, which is the first version of MPEG, is defining a bit stream for digital video and audio. The main application of MPEG-1 is the storage of audio and video data on digital storage media such as CD-ROM. The target bit rate is 1.5 Mbps. The MPEG-1 video coders allow CIF and SIF formats

as input video sequences. To achieve the target bit rate, MPEG-1 adopts DCT, quantization and entropy coders for intra frame coding, and motion compensation for inter frame coding. DCT and ME/MC are used to remove spatial redundancy and temporal redundancy efficiently, respectively. In MPEG video coding, a sequence is divided into a *group of pictures* (GOP). The one GOP is defined from the I-frame to the P-frame, which is the previous frame of the next I-frame.

Based on the ME direction, three different picture types can exist in a GOP. The I-picture (or I-frame) is coded without ME. The P-picture (or P-frame) is coded using one-directional ME from the previous frame. The B-picture (or B-frame) is coded using ME from both previous and future frames. Each frame is divided into slice levels, and each slice consists of several MBs. While all MBs in an I-frame are coded using intra mode, MBs in P- and B-frames can be coded using either intra or inter mode.

In I-frame coding, each block in a MB is transformed with 2D FDCT, while the difference between the current block and the MC block is transformed with 2D FDCT in P-,B-frame coding. After DCT, there are DC and AC components of each MB. Each DC and AC coefficient is quantized differently based on the coding mode. The DC coefficient of intra MB is quantized to 8, 9 or 10 bits. The AC coefficients are quantized using a quantizer weighting matrix. For efficient quantization an intra quantizer weighting matrix and a non-intra quantizer weighting matrix are defined. In MPEG-1, the MVs and quantized DC coefficients are coded using DPCM. The other coefficients are coded using a VLC coding method. For efficient VLC coding, the MPEG standard defines several VLC tables.

2.3.5 MPEG-2

The MPEG-2 is a standard for the generic coding of audio and video data. The MPEG-2 video coding standard is similar to the previous MPEG-1 with extension of some functions to support efficient coding for a wide range of application. The permitted data rate of the MPEG-2 video is up to 100 Mbps. The target applications of the MPEG-2 are storage and transmission of movie quality video and audio data. The MPEG-2 video is widely used for the transmission of video data over satellite, cable, and other broadcast channels. Currently most broadcasting systems such as digital TV, DVD, and HDTV use MPEG-2 to encode video data. To meet the requirements of a large range of applications, the MPEG-2 standard defines a set of profiles and levels. A profile specifies a set of coding features, while a level specifies the spatial and temporal resolutions of the application.

MPEG-2 video is not optimized for very low bit rate coding, but it supports different formats of video sequences. The MPEG-2 video supports interlaced video signal which is the standard of analog TV as well as progressive video signals. Supporting interlaced video is a major difference from MPEG-1 video. Also it supports various chrominance sampling ratios such as 4:2:0, 4:2:2, and 4:4:4. Like MPEG-1 video, MPEG-2 video uses different frame modes for intra and inter coding (I-, P-, and B-frame). Some important features of MPEG-2 video are explained in the following subsections.

Field/frame Prediction

Interlaced video signals consist of two fields per frame, which are the odd field and even field. A MB can be represented with different modes: field mode and frame

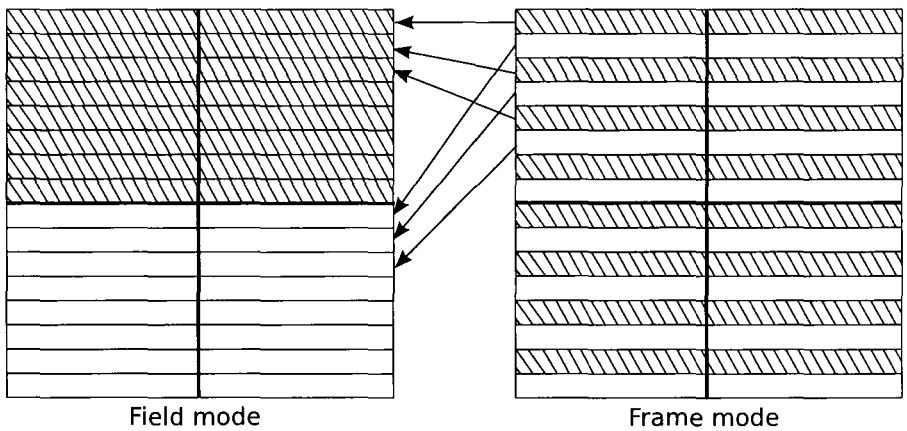


Figure 2.6: Field and frame mode of a luminance macroblock for interlaced video.

mode, which is shown in Figure 2.6. In field mode, each field can be predicted from either field of the previous reference frame.

Field/frame DCT Coding

For interlaced video signals, field DCT can be selected at MB level. The field DCT contributes to better coding efficiency when the difference between consecutive fields is large. For the field DCT, fields are grouped together as shown in Figure 2.6.

Downloadable Quantization Matrix & Alternative Scan Mode

In the MPEG-2 video, unlike with H.261 and MPEG-1, the quantization matrix can be modified for every frame. This option contributes greatly to improve coding performance when a video sequence has dynamic motion. The encoder signals the uses of the new quantization matrix to the decoder by setting the load flag of the bitstream. To scan quantized coefficients, both zigzag and alternative scan orders

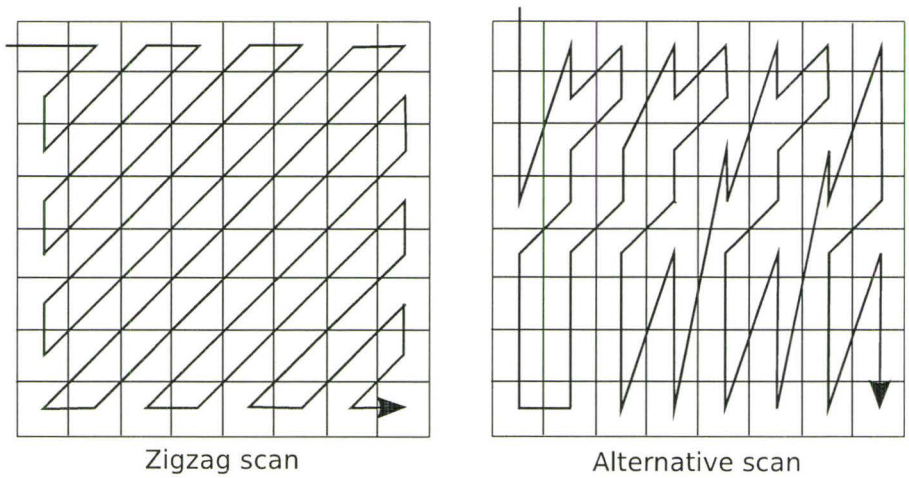


Figure 2.7: Example of zigzag scan and alternative scan ordering.

are used in the MPEG-2 video encoder. Alternative scan order is useful when the interlaced video signal is coded and nonzero quantized coefficients exist at the bottom area of a block. Figure 2.7 shows zigzag scan order and alternative scan order. Note that only zigzag scan order is supported in MPEG-1.

Scalability

A new feature in MPEG-2 video coding is scalability. Using scalability MPEG-2 video codes a video sequence into two or more layers. The MPEG-2 video standard defines four different layered coding modes: spatial scalability, temporal scalability, signal-to-noise-ratio (SNR) scalability, and data partitioning.

Spatial Scalability: Spatial scalability allows coding a video frame with different resolutions. The coding layer with smaller frame resolution is called the base layer. The coding layer with larger frame resolution is called the enhancement layer. The frame of the enhancement layer can be obtained by up sampling the base layer frame.

While the base layer codes a video sequence using either the MPEG-1 or MPEG-2 video encoder, the enhancement layer is coded using only the MPEG-2 video encoder.

Temporal Scalability: Temporal scalability allows coded video data with different frame rates. The spatial resolutions of the two layers are the same. The enhancement layer has higher frame rates than the base layer. The MPEG-2 video decoder can decode only the base layer without the enhancement layer when channel bandwidth is insufficient.

SNR Scalability: SNR scalability is a tool for providing a different quality of video coded data. The spatial resolutions of base and enhancement layers are the same. The better quality of the enhancement layer is achieved by using a finer quantizer.

Data Partitioning: For data partitioning, the parameters of the video encoder are split into a high-priority part and low-priority part. The parameters of high-priority are coded with a different coding scheme from the coding scheme for low-priority parameters.

2.3.6 MPEG-4

The main goal of MPEG-4 visual is providing tools allowing content-based coding for storage, transmission and interactive communication for visual and other elements which are part of multimedia. The MPEG-4 uses the term visual instead of video. The visual data includes both natural and synthetic contents. While other video coding standards employ frame based coding, MPEG-4 visual employs an object based coding. For object information, MPEG-4 uses shape information, which is either binary or grey format. For texture coding, the MPEG-4 visual coder uses a wavelet based transform. The major tools of MPEG-4 are: motion compensation based inter

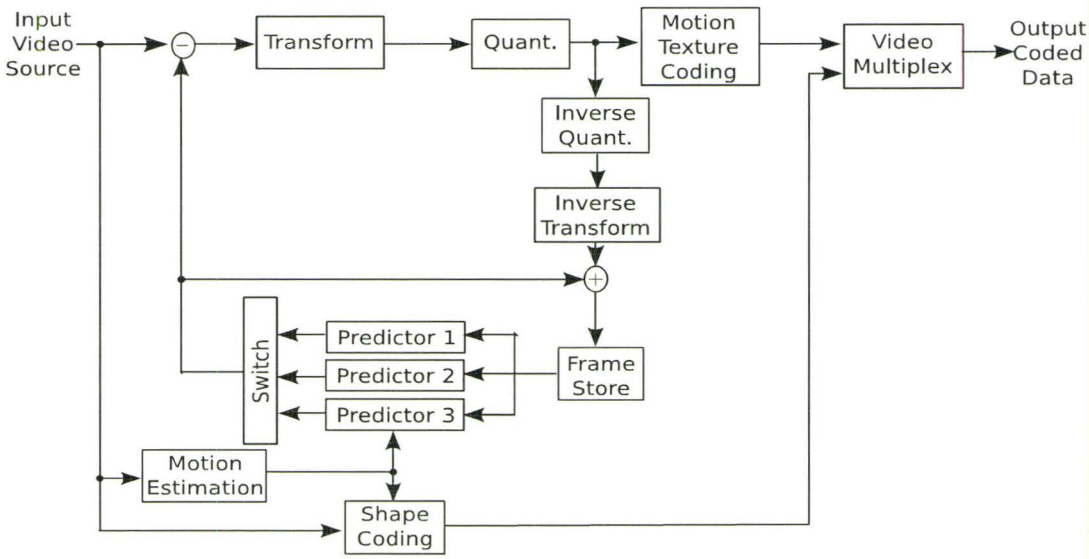


Figure 2.8: The structure of the MPEG-4 encoder.

coding, texture coding, shape coding, sprite coding, interlaced video coding, wavelet based texture coding, error resilience, and scalability. Figure 2.8 shows the encoder structure of MPEG-4. In the following section these tools are investigated in detail.

Motion Compensation Based Inter Coding

Like other video coding standards, MPEG-4 visual encodes an object in an inter frame using ME/MC. MPEG-4 visual coding provides advanced ME tools such as adaptive selection of block size and overlapped ME for the luminance block. In adaptive block size, the ME can be performed based on either 16×16 MB level or four 8×8 blocks level. If four 8×8 blocks level is chosen for ME, four MVs are available to represent one MB. While adaptive block size ME has better performance in low bit rate coding, the overlapped MC reduces the prediction noise.

Texture Coding

A new term, *video object plane* (VOP), is defined for the video object in a frame. I-, P-, and B-VOP are used like I-, P-, and B-frame in other standards. Texture coding uses 2D FDCT to encode I-VOP or the prediction errors of P- or B-VOP. For I-VOP texture coding, intra prediction predicts DC and AC coefficients. The DC and AC predictions are performed using the neighboring blocks of the current block. Also texture coding supports arbitrary shaped object coding. Depending on the position of an object, a MB is classified into one of three types: (1) A MB is defined as an opaque MB if the MB is completely located inside of VOP. (2) If whole points of a MB are located at the outside of VOP, it is called a transparent MB. (3) Otherwise a MB is called a boundary MB. The opaque MB is coded with conventional coding techniques. The transparent MB is not coded. Padding and shape adaptive DCT are used to encode the boundary MB.

Shape Coding

The shape information for an object is called the alpha plane. There are two different alpha planes: the binary alpha plane and gray scale alpha plane. The binary alpha plane is coded with content based arithmetic coding while the gray scale alpha plane is coded with motion compensated FDCT coding. For each shape MB in P-, and B-VOP, there are MVs. Different modes are defined to code shape MBs such as transparent, opaque, intra, inter with shape MVs, and inter without shape MVs.

Sprite Coding

A sprite is a video object existing throughout an entire video sequence. A background in a video sequence is a good example of a sprite. In a sprite coding, first, the object is transmitted to the receiver. Both transmitter and receiver keep the same sprite object. Then only the camera parameters corresponding to the object of each frame are enough to transmit to the receiver instead of all information about the object. The chrominance components of a sprite object are coded with the same method as one the luminance components of the sprite object. Sprite coding contributes to lower the required bit rate significantly.

Interlaced Video Coding

As explained in Section 2.3.5, interlaced video consists of an odd field and even field. In MPEG-4 visual coding, interlaced video coding is based on VOP while it is based on frame in MPEG-2. Two MVs exist for each MB in the interlaced video coding. Shape information is also considered in the interlaced video coding when a static background persists for a long time.

Wavelet Based Texture Coding

Since wavelet based transform provides high coding efficiency and good scalability MPEG-4 visual coding supports wavelet based coding for still images or textures. The transformed coefficients of each band are coded with different quantizers from the coefficients of other bands. Thus the maximum number of quantizers is the same as the maximum number of bands. The MPEG-4 visual coder decides the quantizer of each band.

Error Resilience

MPEG-4 visual coding supports error resilience tools to recover and transmit coded data over error prone transmission channels when channel error exists. The error resilience tools include resynchronizations, data partitioning, and error concealment. Resynchronizations periodically insert synchronization marks. To minimize lost data, reversible variable length codes are employed. Error concealment is implemented at the decoder side because errors occur during transmission.

Scalability

In the MPEG-4, object based temporal and spatial scalabilities are provided. Thus, an enhancement layer can be applied to only an object or area of a frame. The basic concept of scalability is the same as that of MPEG-2, which is described in Section 2.3.5.

2.3.7 H.264/AVC

The H.264/*advanced video coding* (AVC) is the latest video coding standard. The H.264/AVC has significantly improved coding efficiency compared to conventional video coding standards such as H.263 and MPEG-4. Like other video coding standards, the H.264/AVC encoder uses a DCT and ME/MC based hybrid coding scheme as shown in Figure 2.9. To improve coding efficiency, H.264/AVC employs several new techniques such as intra prediction, ME using multiple reference frames and variable block sizes, integer transform coding, adaptive de-blocking filter, encoder control based on rate control, and advanced entropy coding schemes. These tools are described in the following subsections in detail. More details about H.264/AVC *video*

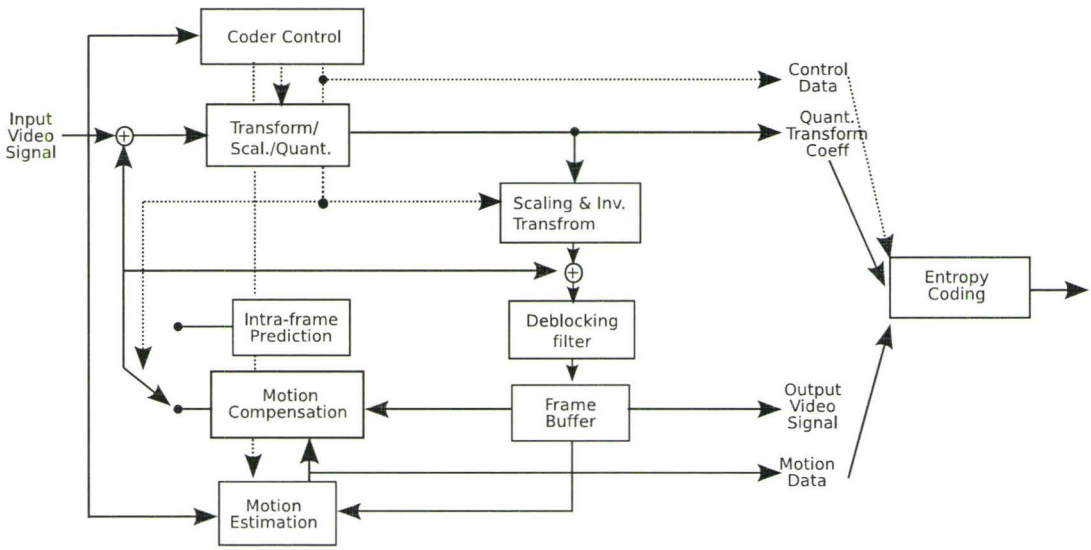


Figure 2.9: The structure of the H.264/AVC video encoder.

coding layer (VCL), *network abstraction layer*, and overall structure are explained in [46].

Intra Prediction

Intra prediction allows that intra MB can be coded with prediction using already coded neighboring MBs in the same frame. The intra prediction is only allowed for luminance blocks. Both 16×16 MB and sixteen 4×4 blocks are predictable. The former is called INTRA_16x16 and the latter is called INTRA_4x4. While only one prediction mode exists for INTRA_16x16 mode, sixteen different predictions are possible in INTRA_4x4 mode. In INTRA_16x16 mode, four different predictions are provided which are vertical prediction, horizontal prediction, DC prediction, and plane prediction. In INTRA_4x4 mode, nine different predictions exist, where one

prediction is for DC prediction and the other eight predictions indicate the direction of prediction for each pixel in each 4×4 sized block. More details about each prediction mode can be found in [47]. The intra prediction improves coding efficiency when the input video sequence has smooth areas in a MB.

Motion Estimation and Compensation

H.264/AVC also employs ME and MC to increase coding efficiency and to reduce the required bit rate by removing temporal redundancy [46]. To improve coding efficiency, the ME of H.264/AVC video employs various advanced techniques such as multiple reference frames, quarter-pixel MC, and variable block size ME. Supporting both variable block sizes and multiple reference frames improves coding efficiency significantly but it also increases the computational complexity of the encoder. MB mode can be one of 16x16, 16x8, 8x16, 8x8, 8x4, 4x8, and 4x4, as shown in Figure 2.10. After ME, inter skip mode (P_SKIP) is used if the following conditions are satisfied: 1) the quantized block has zero pattern block, 2) the MV difference is zero, and the reference frame is the previous one. Neither MV nor residue are transmitted to the decoder when a MB is coded with P_SKIP mode.

By considering the Lagrangian cost function using both distortion and rate, the best MB mode, which has the minimum value of the cost function, is chosen to represent the MB. The detail of the Lagrangian cost function is explained in Section 2.3.7. For quarter-pixel ME, a reference frame is generated by interpolation using a one dimensional 6-tap FIR filter and averaging for half-pixel and quarter-pixel positions, respectively. The H.264/AVC video supports multiple reference frames based ME/MC to achieve high coding performance.

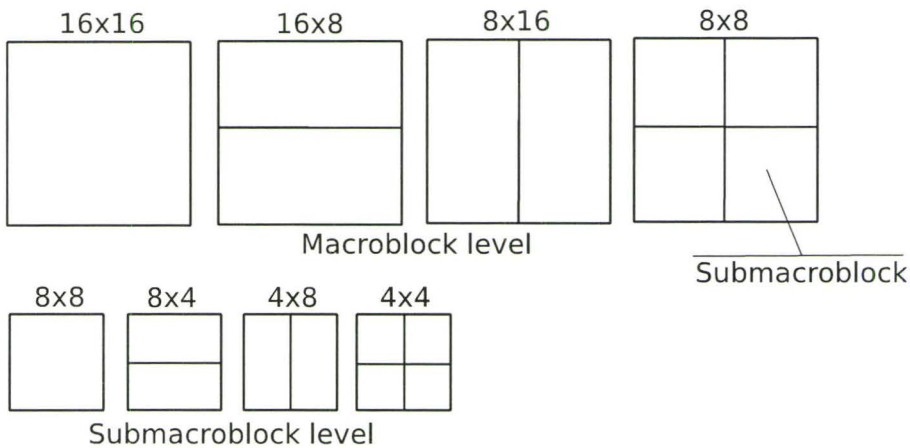


Figure 2.10: Partitioning of a macroblock and a submacroblock.

Integer Transform Coding

Conventional video coding standards use 8x8 block based 2D FDCT to remove spatial correlation. The H.264/AVC employs integer transform instead of FDCT. The block size can be as small as 4x4 or 2x2. Integer transformation matrices are defined as

$$H_1 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}, H_2 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}, \text{ and } H_3 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (2.8)$$

The transformation matrix H_1 is applied to all prediction errors of luminance and chrominance components. For INTRA_16x16 MB both H_1 and H_2 transformation matrices are used together. The H_2 transformation matrix is used for sixteen DC coefficients after transformation with the H_1 transformation matrix. The H_3 transformation matrix is used for four DC coefficients of chrominance block after

transformation with the H_1 transformation matrix. The integer transform coding reduces the computational complexity of the encoder because it requires only shift, add and subtract operations. Also by using exact integer operations in the inverse transform, the inverse transform mismatches are avoided [46]. After integer transformation, quantization is performed with one of 52 quantization parameters.

Adaptive De-blocking Filter

All block based video coding schemes produce blocking artifacts on the reconstructed video frame. By removing block edges using post-filtering, higher visual quality can be achieved. In H.264/AVC, the filtering process is inserted as part of the encoder to remove the blocking artifacts of the reference frame before ME/MC. The filter is called the de-blocking filter. Since blocking artifacts are caused mainly by quantization the filtering parameters are related to quantization parameters. In the H.264/AVC encoder, de-blocking is adaptive on slice, block edge, and sample levels. All filtering processes are involved in only additions and shifts to minimize the complexity. Usually de-blocking filtering improves the subjective quality of the reconstructed frame.

Encoder Control Based on Rate Control

In terms of the rate and distortion of lossy video coding techniques, the ideal is minimizing distortion while keeping or reducing bit rates, or decreasing bit rates while keeping or decreasing distortion. The rate distortion theory says that rate is in inverse proportion to distortion i.e., increasing rates decreases distortion. All video encoder and decoder pairs provide rate distortion curves. For specific applications, the user chooses one specific point on the rate distortion curve to set the encoding

parameters. The H.264/AVC video encoder tries to minimize both rate and distortion at MB level. The Lagrange optimization algorithm is introduced to establish a cost function J which is

$$J(I_k|\lambda_k, QP) = D(I_k|QP) + \lambda_k R(I_k|QP) \tag{2.9}$$

where I_k is one of the seven modes ($k \in 1, 2, \cdots, 7$) from Figure 2.10, QP is a quantization parameter and $D()$ and $R()$ are distortion and rate functions, respectively, and λ_k is the Lagrangian multiplier. Distortion function $D()$ can be a function given by Equation 2.1. The I_k minimizing cost function J is selected according to the type of the MB. For each MB, calculating the cost function J is computationally intensive.

Advanced Entropy Coding

H.264/AVC provides two entropy coding methods: a *context adaptive variable length coding* (CAVLC) and a *context based adaptive binary adaptive coding* (CABAC). Both coders improve the coding efficiency of the H.264/AVC video encoder compared to conventional fixed *variable length codes* (VLCs), which are adopted in other video coding standards. Due to its low computational complexity the baseline of H.264/AVC uses only CAVLC. In CAVLC entropy coding mode, 32 different VLCs are defined and used. Although its computational complexity is intensive, CABAC is used when significantly improved coding efficiency is required. The structure of CABAC includes: binarization, context modeling, and binary arithmetic coding. More details about CABAC can be found in [48].

2.4 Performance Parameters and Comparison

2.4.1 Distortion Criteria

While lossless coding and decoding reproduces an identical image to the original image, there is a difference between original and reconstructed images in lossy coding. Quantifying the difference between original and reconstructed images is not easy because of the difficulty of modelling human visual perception. Two popular measures of distortion between the original and reconstructed images are squared error and absolute difference error. These are called difference distortion measures. Assume x is the original source and y is the reconstructed one, then the squared error is defined by [49]

$$d_S(x, y) = (x - y)^2 \tag{2.10}$$

and the absolute difference is given by

$$d_A(x, y) = |x - y|. \tag{2.11}$$

Then for the one set of the original source $\{x_1, x_2, x_3, \cdots, x_N\}$ and the reconstructed data $\{y_1, y_2, y_3, \cdots, y_N\}$, the *mean absolute error* (MAE) is

$$MAE = \frac{1}{N} \sum_{n=1}^{N} |x_n - y_n|, \tag{2.12}$$

and the *mean squared error* (MSE) is

$$MSE^2 = \frac{1}{N} \sum_{n=1}^{N} (x_n - y_n)^2 . \tag{2.13}$$

The *signal to noise ratio* (SNR) is defined as

$$SNR = \frac{ASV^2}{MSE^2} \tag{2.14}$$

where ASV^2 is the average squared value of the original source, $ASV^2 = \frac{1}{N}\sum_{n=1}^{N} x_n^2$. To express the SNR of a very wide dynamic range source, the SNR is usually expressed in terms of a logarithmic decibel scale as

$$SNR(\text{dB}) = 10\log_{10}\frac{ASV^2}{MSE^2}(\text{dB}). \tag{2.15}$$

With the same manner the *peak signal to noise ratio* (PSNR) is defined by

$$PSNR(\text{dB}) = 10\log_{10}\frac{x_{peak}^2}{MSE^2}(\text{dB}) \tag{2.16}$$

where x_{peak} is the biggest value of the original source data.

2.4.2 Bit Rate

Another criterion for the performance of video coding is bit rate. Because the objective of video coding is reducing the required data size, the achieved compression ratio of the video encoder is an important factor. Bit rate is the transmitted amount of a coded video stream over time. Bit rate is typically expressed as kilobits per second (kbps) or Megabits per second (Mbps). The bit rate of the video coder depends on the resolution of the input video sequence, compression ratio, and frame rates. The bit rate R is

$$R = \frac{b \times lines \times pels \times f}{c}[\text{bps}] \tag{2.17}$$

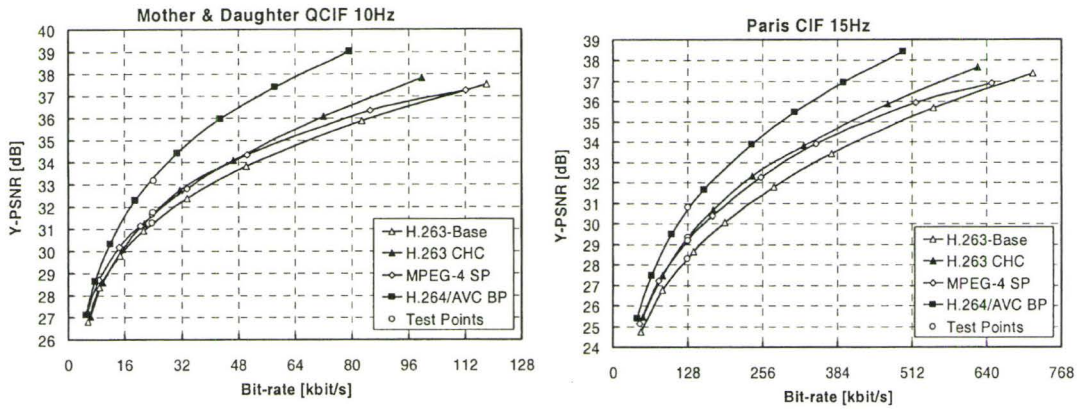
where b is the number of bits per pixel, *lines* is the number of vertical lines of a frame, *pels* is the number of pixels per line, f is the frame rate, and c is the compression ratio. In a given input video source, the bit rate R can be reduced by increasing the compression ratio c.

2.4.3 Comparison of Video Coding Standards

In this section, several video coding standards are compared in terms of rates and distortions. All of the data and graphs used in this section are from [1]. For impartial comparison, Wiegand *et al.* [1] compared four different video coding standards based on three applications. They considered video conferencing applications, video streaming applications, and entertainment quality applications. Four video coding standards, which are MPEG-2, H.263, MPEG-4, and H.264/AVC, are compared each other. The details about each comparison result are in the following subsections.

Video Conferencing Applications

In video conferencing applications, real time video encoding and low target bit rates are the main objective. To meet the target bit rates, four QCIF and four CIF sequences are used. The frame rates are 10 and 15 fps for QCIF, and 15 and 30 fps for CIF, respectively. For low complexity of the encoder, the H.263 baseline and *conversational high compression* (CHC) profiles, the MPEG-4 simple profile, and the H.264/AVC baseline profiles are compared to each other.



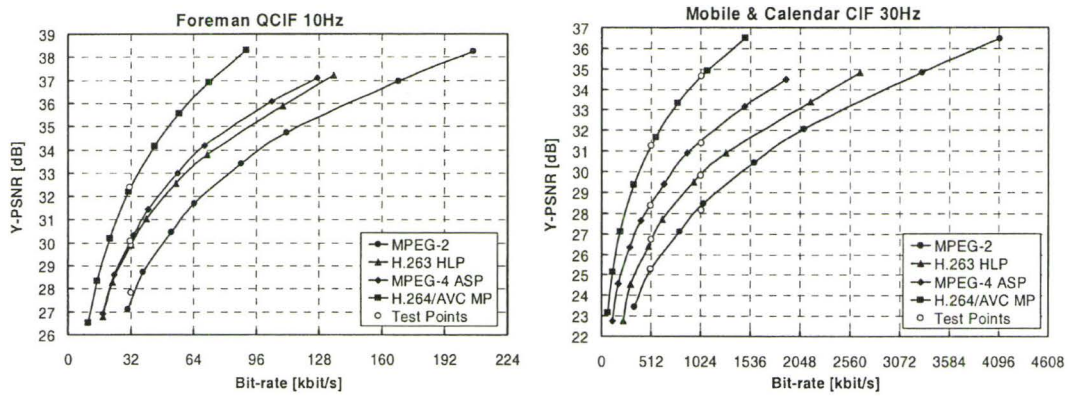
(a) (b)

Figure 2.11: Rate distortion curves for video conferencing applications from [1]: (a) **Mother&daughter** encoded at 10 fps and (b) **Paris** encoded at 15 fps.

Video Streaming Applications

For video streaming applications, four video coding standards use the following profiles and levels: MPEG-2 main profile, H.263 high latency profile, MPEG-4 advanced simple profile, and H.264/AVC main profile. Eleven different sequences having different resolution and frame rates are used as test input sequences. Figure 2.12 (a) and (b) depict the rate distortion curve of the **Foreman** and **Mobile** sequences, respectively. The **Foreman** sequence is QCIF format and the frame rate is 10 fps. The **Mobile** sequence is CIF format and the frame rate is 30 fps. The rate distortion curve in Figure 2.12 shows that H.264/AVC has the best performance among the four video encoders. H.264/AVC improves coding efficiency when the test sequence has complex motion such as the **Mobile** sequence [1].

The average bit rate saving of H.264/AVC is around 70% and 35% compared to MPEG-2 and MPEG-4, respectively. The improved coding efficiency of H.264/AVC



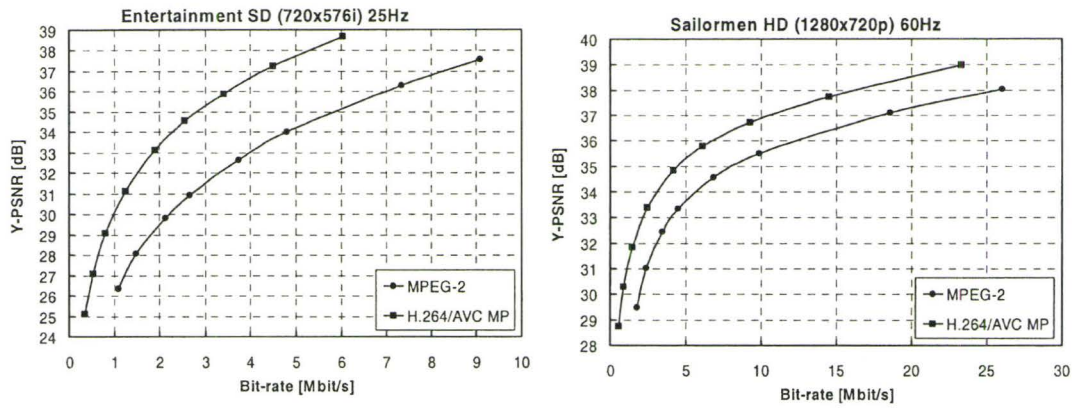
(a) (b)

Figure 2.12: Rate distortion curves for video streaming applications from [1]: (a) **Foreman** encoded at 10 fps and (b) **Mobile** encoded at 30 fps.

is from the flexible motion model using multiple reference frames and variable block sizes ME, de-blocking filtering to remove the blocking effect, and a context-based arithmetic coding scheme.

Entertainment Quality Applications

In entertainment quality applications, to meet required high bit rates, video sequences having more than 720x480 resolution are used. Since H.263 and MPEG-4 are not standards for entertainment quality applications, only MPEG-2 and H.264/AVC are compared to each other. The levels and profiles are: MPEG-2 main level at main profile for standard definition and MPEG-2 high level at main profile for high definition sequences, and H.264/AVC main profile. Figure 2.13 shows the resulting rate distortion curve. Figure 2.13 (a) is the result when the **Entertainment** sequence is used as the test sequence. The **Entertainment** sequence is 720x576 size interlaced



(a) (b)

Figure 2.13: Rate distortion curves for video entertainment quality applications from [1]: (a) **Entertainment** encoded at 25 fps and (b) **Sailormen** encoded at 60 fps.

format and the frame rate is 25 fps. The test sequence of Figure 2.13 (b) is **Salesman** which has 1280x720 size progressive format and 60 fps frame rate. In both test sequences H.264/AVC has better performance than MPEG-2. The average bit rate saving of H.264/AVC is about 55% at low bit rate area and about 35% at high bit rate area.

2.5 Summary

In this chapter, several tools which are widely used for video coding are introduced. The explanation of existing video coding standards follows with key features of each standard. Then the performance parameters are introduced to measure coding efficiency. In video coding standards, rate and distortion are commonly used performance parameters. While the rate is measured in bits per second, the distortion is measured

by the PSNR. For fair comparison, both rate and distortion are depicted on a single graph. The curve on the graph is called the rate distortion curve. The performance of several video coding standards is discussed. In terms of the rate distortion curve, H.264/AVC has the best performance among various video coding standards such as H.263, MPEG-2, and MPEG-4 in different applications.

Chapter 3

Computational Complexity of the Video Encoder

Computational complexity of the video encoder is one of the key points and should be considered in implementing real time applications. Since every block in the video encoder contributes to coding performance the complexity of the video encoder increases in proportion to the coding performance of the video coder. Thus there are trade-offs between complexity and performance. Much research has been done to reduce the encoder complexity while keeping the coding performance. In this chapter the complexity issue of conventional video encoders is discussed. Especially the complexity of ME is explained in detail.

3.1 Introduction

The complexity of the video encoder is important for new emerging applications such as video telephony, multi array cameras, etc. In those applications the battery life

and the execution speed of real time applications are directly related to the complexity of the algorithms used in the functions of a system. The encoder complexity is more important when the encoder resources are strictly limited. Thus designing a low complexity algorithm is mandatory in real time applications. Much research has been done to reduce the encoder complexity. Since video coding adopts many technologies such as ME/MC, transformation/quantization, mode decision and intra prediction, there are many ways to achieve a lower complexity video encoder. For example, *fast discrete cosine transform* (FDCT) is a method of achieving transformation with reduced complexity. Chang and Wang [50] proposed a FDCT based hardware implementation. Hsiao *et al.* [51] developed a DCT hardware implementation without multiplication.

Prediction is a widely used technique for video coding because it removes both spatial and temporal redundancy efficiently. In video coding there are two predictions: (1) intra prediction, which estimates the current block from the current frame, and (2) inter prediction, which estimates the current block from the previous frame(s). While the intra prediction removes spatial redundancy, temporal redundancy is eliminated using inter prediction. For inter prediction ME is employed to find the best approximation of the current block. Since ME adopts a greedy algorithm it has huge computational complexity. Most of the computational complexity of the video encoder can be reduced by designing an efficient ME algorithm.

There are many factors which are related to the complexity of ME. For example, the number of reference frames, search window size, search pattern and initial motion vector all impact the complexity of ME. Multiple reference frames based ME contributes to increased coding performance in many cases. But the complexity of the

video encoder linearly increases in proportion to the number of reference frames.

In this chapter, the computational complexity of the video encoder is discussed in detail. First, the encoder complexity of the current video coding standards is introduced. Especially the computational complexity at the ME block is discussed in detail because ME is responsible for most of encoder complexity. Previous research on topics such as the fast ME algorithm, fast mode decision algorithm and fast reference frame selection algorithm are introduced to help motivate the design a low complexity video encoder.

This chapter is organized as follows: In Section 3.2, the distribution of video encoder complexity is discussed. Especially, the complexity of each block of the H.264/AVC video encoder is compared in this Section. Various ME algorithms for video coding, such as motion search patterns are introduced in Section 3.4. In Section 3.5, mode decision algorithms which were used only in the H.264/AVC video encoder are discussed. The summary of this chapter is provided in Section 3.6.

3.2 Distribution of Encoder Complexity

Most video encoders have a similar structure consisting of prediction error signal, transform/quantization, intra prediction, ME/MC and entropy coding. To design a low complexity video encoder, checking the computational complexity of each function is useful. The computational complexity can be measured in terms of encoding time or the number of processor clock cycles required to perform the function. Due to its simplicity of implementation, the encoding time and ME time are used for encoder complexity in this literature. Figure 3.1 shows the distribution of encoder complexity across different sub-components when the reference software of H.264/AVC encoder

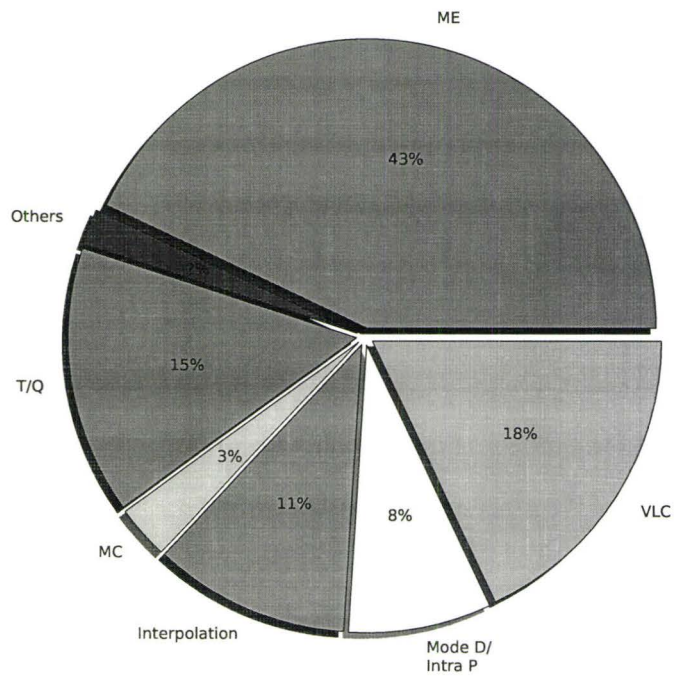


Figure 3.1: The encoder complexity of the H.264/AVC video encoder.

is used. As we can see from Figure 3.1, the encoder complexity is as follows: the ME and MC take around 50% of encoder complexity, the portion of mode decision and intra prediction is 8%, while other functions take around 44% of encoder complexity. The ME was performed with one reference frame. As the number of reference frames increases, the portion of ME/MC increases.

In H.264/AVC, two alternative entropy coding methods are specified. *Context adaptive variable length code* (CAVLC) performs entropy coding with low complexity. *Context based adaptive binary arithmetic coding* (CABAC) has better coding performance than CAVLC but it has greater complexity. Both CAVLC and CABAC

algorithms represent big improvements in terms of coding efficiency compared to the conventional *variable length coding* (VLC). For several testing sequences, CAVLC reduces bit rate by 2% to 7% compared to conventional VLC based on a single Exp-Golomb code. Also CABAC reduces the required bit rate by 5% to 15% compared to conventional VLC. Both the CAVLC and CABAC have more computational complexity. For example, CAVLC takes around 18% of encoder complexity as we can see from Figure 3.1.

Similar to the previous video coding standards, H.264/AVC employs two dimensional DCT to reduce the spatial redundancy of the predicted video signal. While the previous coding standards apply 2D DCT to the 8×8 size block, the block size of DCT is mainly 4×4 and can be as small as 2×2 in the H.264/AVC. As mentioned in Section 2.3.7 the H.264/AVC defined three different types of transformations. Those three transformations are called integer Hadamard transformations.

Since the Hadamard transformations have only integer numbers ranging from -2 to 2, computing the transformation and inverse transformation have low complexity. The integer Hadamard transformations can be performed only using shift, add, and subtract operations. Also the integer transformation avoids the mismatches in transformation and inverse transformation. After transformation all coefficients are quantized by a quantization formula. For various qualities and compression ratios 52 *quantization parameters* (QPs) are defined. Both transformation and quantization blocks take around 15% of encoder complexity in the H.264/AVC video encoder.

Mode decision and intra prediction are other features increasing coding performance in the H.264/AVC video coding system. Figure 3.2 depicts the concept of multiple reference frames and variable block modes for each MB. Two intra modes

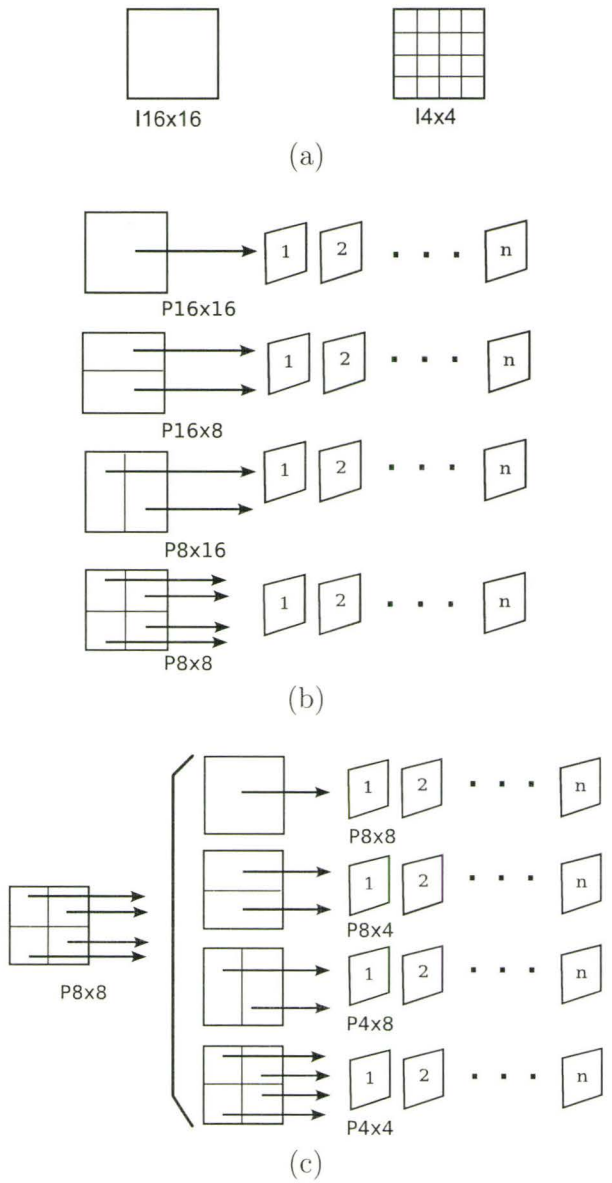


Figure 3.2: Multiple reference frames motion estimation with various sub macroblock partitions: (a) intra mode, (b) inter mode with macroblock partitions and (c) inter mode with sub macroblock partitions mode.

such as I16 × 16 and I4 × 4 are available to represent a MB as shown in Figure 3.2(a). The inter mode block mode can be one of P16 × 16, P16 × 8, P8 × 16 as shown in Figure 3.2(b) and P8 × 8, P8 × 4, P4 × 8, and P4 × 4 as shown in Figure 3.2(c). In addition, in many instances, a mode known as skip is available. By considering the Lagrangian cost function using both distortion and rate, the best mode is chosen for each MB.

The Lagrangian cost function for a macroblock MB is

$$J(MB, I|\lambda_{mode}, QP) = D(MB, I|QP) + \lambda_{mode} R(MB, I|QP) \tag{3.1}$$

where I represents one of the possible block modes as shown in Figure 3.2, QP is a quantization parameter and $D()$ and $R()$ are a distortion and rate function, respectively. λ_{mode} is a Lagrange multiplier. The I that minimizes cost function J is selected as the block mode for MB. But, finding the best block mode I is computationally heavy. In our simulation, mode decision and intra prediction take around 8% of encoder complexity.

3.3 Fast DCT for Low Complexity Video Encoder

DCT is the most widely used transformation in video coding because it has satisfactory energy compaction for a video signal. Performing DCT requires floating point multiplications, making it slow in both hardware and software implementation. Scaling and approximation of a floating point number into an integer number are used to achieve a fast DCT implementation [52], [53]. Cham [54] developed integer DCT

by searching integer orthogonal transforms with symmetry and similar energy compaction.

Although integer DCT provides a fast algorithm, the complexity of DCT is still high because it requires many multiplications. Liang and Tran [55] used the lifting scheme to implement the approximated DCT function called the binDCT without multiplication. Kutka [56] achieved fast DCT by replacing multiplications with a look-up table. In contrast to the previous video coding standards which mainly use 8×8 DCT, H.264/AVC uses mostly a 4×4 block based transform for the residual signal. It is well known that a larger transform has better performance when most of the energy is contained in the low frequency components. The smaller transform has significant advantage when prediction is accurate. In H.264/AVC, the 4×4 integer DCT is defined as

$$\begin{aligned} E(u,v) &= \sum_{x=0}^{3} \sum_{y=0}^{3} e(x,y) \cdot A(x,u) \cdot A(y,v) \\ A(m,n) &\triangleq \left\langle \frac{2.5C(n)}{\sqrt{2}} \cos \frac{(2m+1)n\pi}{8} \right\rangle \end{aligned} \tag{3.2}$$

where $e(x,y)$, $0 \leq x, y \leq 3$, is a residual signal; $C(n) = \frac{1}{\sqrt{2}}$ for $n = 0$, and $C(n) = 1$, otherwise. The operator $\langle x \rangle$ denotes rounding x to the nearest integer.

If a MB is encoded with I16 $\times$ 16 mode, each 4×4 residue block is transformed using Equation 3.2. Then the Hadamard matrix is applied to transform the DC coefficients of each 4×4 block as

$$T_Y = \frac{1}{2} H_2 E H_2^{\mathrm{T}} \tag{3.3}$$

where H_2 is defined in Equation 2.8. For the chrominance block, there are four 4×4 blocks per MB. The transformation for the DC coefficient of the chrominance block

is performed using a 2×2 Hadmard transform as

$$T_C = H_3 E H_3^{\mathrm{T}} \tag{3.4}$$

where E is the matrix of grouped DC coefficients of the chrominance blocks and H_3 is a Hadamard matrix defined in Equation 2.8.

In H.264/AVC, if the Hadamard transform is enabled for a residual block after inter- or intra-prediction, the *sum of absolute transformed difference* (SATD) is used as a distortion measure. The SATD can be calculated as

$$\mathrm{SATD} = \frac{1}{2} \sum_{u=0}^{3} \sum_{v=0}^{3} |B(u,v)| \tag{3.5}$$

where B is the 4×4 Hadamard transformed residue block given by

$$\begin{aligned} B &= H_2 E H_2^{\mathrm{T}} \\ &= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} e_{00} & e_{01} & e_{02} & e_{03} \\ e_{10} & e_{11} & e_{12} & e_{13} \\ e_{20} & e_{21} & e_{22} & e_{23} \\ e_{30} & e_{31} & e_{32} & e_{33} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}^{\mathrm{T}} \end{aligned} \tag{3.6}$$

where E is a 4×4 residual block.

Since SAD is not available before DCT and quantization, SATD information can be used when the Hadamard transformation is enabled for H.264/AVC. Wang and Kwong [57] used SATD to predict the zero quantized DCT coefficients. The predicted zero quantized coefficients are used for reducing the computational complexity of DCT and the quantization block. A hardware implementation of fast DCT can be found

in [51].

3.4 Motion Estimation Algorithms for Low Complexity

Motion estimation is a key function in image processing and inter picture video coding and has been utilized for various applications involving single image and image sequence, particularly since the early 1990's [58], [59] and [60]. Examples include algorithms that have been developed for medical imaging [61], stereo imaging for robotics [62], and model based facial image compression [63]. More important, all existing video coding standards adopt ME and MC by exploiting temporal redundancy, a pervasive characteristic of video sequences [47], [64].

H.264/AVC, which is the latest video coding standard, also employs ME/MC to increase coding efficiency and to reduce the required bit rate [46]. To improve coding efficiency, the H.264/AVC employs various advanced techniques such as multiple reference frames, intra prediction, quarter-pixel MC, 4×4 integer DCT and variable block size ME. Among these functional blocks, ME is the most computationally complex part of the encoder. The main goal of ME is to find the best approximation for each block of a frame (known as a MB) from a reconstructed previous frame (or reference frame). Supporting both variable block sizes and multiple reference frames improves coding efficiency at the expense of increasing the computational complexity.

Although there is increased computational complexity, the use of multiple reference frames based ME has the advantage over single reference frame ME in cases such as:

(1) where the objects have repetitive motion with a period of more than one frame,

(2) there are new visible areas of an object and/or uncovered background that are not easy to approximate from a single previous frame,

(3) there are sources of noise in the previous frame,

(4) the motion vector of a MB is large.

In these cases, using multiple reference frames for ME improves the prediction of motion for the current object. The advantage of multiple reference frames based ME is also discussed in [21], [22] and [23].

Reducing ME time is critical in real time image processing and video coding applications. In the H.264/AVC encoder for example, depending on the number of reference frames and the search range, approximately 50% to 70% of the complexity can be attributed to ME [65]. To reduce the complexity of the encoder, the design of efficient ME is an open, challenging problem. Better ME generally results in a lower bit rate and thus many fast ME algorithms have been proposed to reduce complexity while maintaining coding efficiency [65], [66].

3.4.1 The Computational Complexity of Motion Estimation

To find the best approximation for each MB in the search window of a reference frame, the ME comprises both calculating and comparing rate and distortion for every search point. Due to the low complexity, *sum of absolute difference* (SAD) is usually employed for block distortion measurement. For a block, at position (x, y),

the SAD is

$$\begin{aligned} & SAD_{(x,y)}(mv_x, mv_y) \\ & = \sum_{i=1}^{N} \sum_{j=1}^{N} |f_t(x+i, y+j) \\ & \quad - \hat{f}_{t-\Delta t}(x+i+mv_x, y+j+mv_y)| \end{aligned} \tag{3.7}$$

where f_t is the frame at time t, $\hat{f}_{t-\Delta t}$ is the reconstructed reference frame at time $t-\Delta t$, N is the block size, and mv_x, mv_y are MVs of horizontal and vertical direction, respectively. Note that $-w \leq mv_x, mv_y \leq +w$, where w is the search window size.

Since the greatest contributor to the complexity of the image and video processing is from ME, it is useful to characterize the computational complexity of ME. Assuming ME is done using SAD as the block distortion measurement, then for each image frame consisting of M MBs, the complexity for each frame C_{frame} can be written as

$$\mathcal{C}_{frame} = M \times R \times I \times \mathcal{C}_{block} \tag{3.8}$$

where R is the number of reference frames and I is the number of MB modes. The complexity function for each MB $\mathcal{C}_{block}$ is given by

$$\mathcal{C}_{block} = S \times \mathcal{C}_{SAD} \tag{3.9}$$

where S represents the number of search points and $\mathcal{C}_{SAD}$ is the computational complexity to calculate the SAD of the MB.

The complexity of ME is thus directly related to the parameters S, R, and I. Given search window size w, the number of reference frames and the number of possible MB modes, reducing S or R or I will reduce the computational complexity of

ME.

3.4.2 Search Pattern Algorithms (Reducing S)

When a full search pattern is employed, the number of search points is $S = (2w_h + 1) \times (2w_v + 1)$, where w_h and w_v are the search window size in the horizontal and vertical directions, respectively. By calculating the block distortion measurement of the specific points on each search pattern instead of all points in the search window, the computation of ME is effectively reduced. Several algorithms for efficient search patterns have been proposed to reduce S such as in [4], [5] and [6].

More efficient search pattern algorithms can be found in [10] and [15]. Four of the most popular search patterns are depicted in Figure 3.3. The computational complexity of ME can also be reduced using an adaptable search window size as is proposed in [35] and [36].

3.4.3 Reference Frame Selection Algorithm

In H.264/AVC ME is allowed to search MV for current MB from multiple reference frames. In the multiple reference frames based ME, a MB in the current frame is predicted from one reference frame out of a large number of previously decoded frames. The required computational complexity of H.264/AVC is highly increased in proportion to the number of searched reference frames. Figure 3.4 depicts the scenario of the multiple reference frames based ME procedure. In H.264/AVC, ME searches a MV for the current MB from the reference frame and returns the MV (mv_x, mv_y)

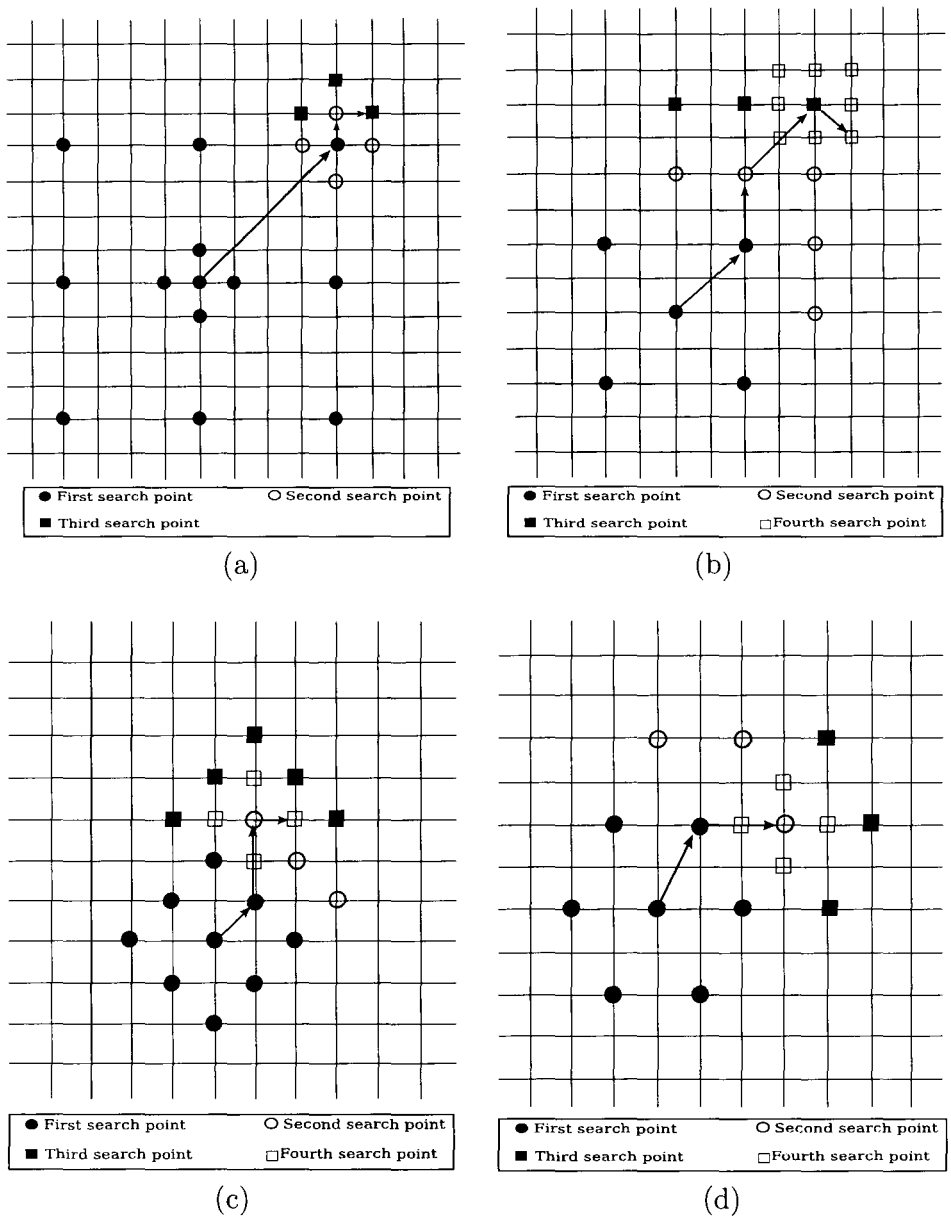


Figure 3.3: Four examples of search patterns: (a) three step search, (b) four step search, (c) diamond search and (d) hexagon search.

minimizing the cost function which is

$$J(\overrightarrow{\mathrm{mv}}, \lambda_{\mathrm{motion}}) = D\left(s, c(\overrightarrow{\mathrm{mv}})\right) + \lambda_{\mathrm{motion}} R\left(\overrightarrow{\mathrm{mv}} - \overrightarrow{\mathrm{mv}_p}\right) \tag{3.10}$$

where $\overrightarrow{\mathrm{mv}_p}$ is a predicted MV and $\lambda_{\mathrm{motion}}$ is a Lagrange multiplier, and the $\overrightarrow{\mathrm{mv}}$ is a MV with horizontal and vertical directions (mv_x, mv_y).

Although the H.264/AVC supports both exhaustive motion search and fast motion search algorithms, the MV search procedure is the most computationally intensive part in the H.264/AVC video encoder. For multiple reference frames based ME, the same search process is applied to each reference frame. Thus, the amount of computational complexity is high. Su and Sun [23] mentioned that multiple reference frames based ME achieves better prediction when the test sequence has the following characteristics:

(1) Repetitive motion.

(2) An uncovered background.

(3) Alternating camera angles that switch back and forth between two different scenes.

(4) Object movement with a non-integer pixel displacement.

(5) Shadow and lighting changes.

(6) A shaking camera.

(7) Noises in the source video signal produced by the camera and other factors.

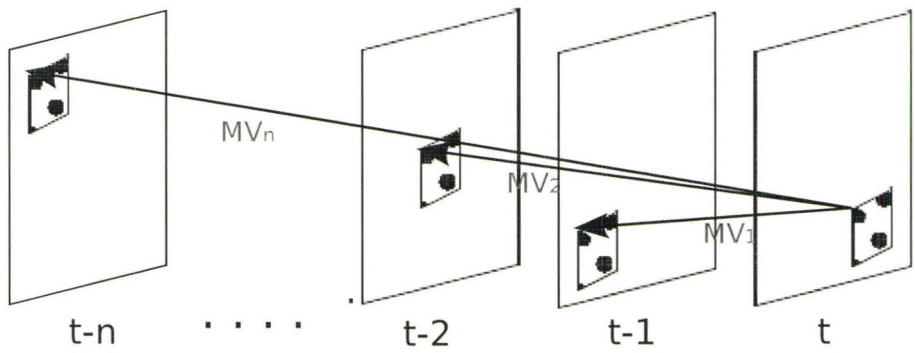


Figure 3.4: Motion estimation based on multiple reference frames.

Multiple reference frames based ME has better performance than single reference frame based ME at the high bit rate position [23]. Although there are improvements on coding performance, multiple reference frames based ME wastes encoder resources without any advantage in many video test sequences. Huang *et al.* [67] showed that 80% of the optimal MVs determined by the reference software are from the nearest reference frame. There are many approaches to reduce the number of reference frames.

Xu and Xiao [68] used spatial correlation to reduce the number of reference frames for block modes and removed unnecessary sub-block modes. Ozbek and Tekalp [69] reduced 23% of encoding time with similar quality and bit rate by selecting reference frames using the histogram similarity based method. Huang *et al.* [65] proposed a context based adaptive method to speed up the multiple reference frames based ME. To achieve 30% to 80% reduction of ME computation, they analyzed the available information after intra prediction and ME from the previous reference frame.

3.5 Mode Decision Algorithm for H.264/AVC

The reference software of H.264/AVC recommends two kinds of mode decision algorithms which are low and high complexity mode decision. The current H.264/AVC reference software performs ME and mode decision based on *rate distortion optimization* (RDO). RDO is performed based on the Lagrangian cost function defined in Equation 3.1. At first, ME evaluates the *rate distortion* (RD) cost for all block types of inter mode. After that the mode decision compares the RD cost of each inter mode and each intra mode. The mode having the minimal RD cost is selected to represent the current MB. It is time consuming work calculating and comparing the RD cost of all modes. Figure 3.2 shows all the possible modes in the P-frame. In Figure 3.2(a) two intra modes are depicted. For intra mode I16 $\times$ 16, four interpolation directions must be checked, and for I4 $\times$ 4, nine interpolation directions must be checked.

Figure 3.2(b) shows the different block sizes of inter mode which can be 16 $\times$ 16, 16 $\times$ 8, 8 $\times$ 16 or 8 $\times$ 8, and each 8 $\times$ 8 size block can be further split into sub-blocks. The sub-block size can be 8 $\times$ 8, 8 $\times$ 4, 4 $\times$ 8 or 4 $\times$ 4 as shown in Figure 3.2(c). ME examines all the modes to decide the optimal mode for a MB. Reducing the computational complexity of mode decision has been the most challenging research topic and much work has been done to reduce the complexity of mode decision in H.264/AVC [32], [27] and [26]. Fast inter and intra mode decision algorithms are summarized in the following subsections.

3.5.1 Intra Mode Decision Algorithm

In H.26/AVC, various intra modes are specified as well as inter mode. Intra prediction is performed in the pixel domain using the neighboring pixel values of previously

coded blocks. For intra prediction, I16 × 16 and I4 × 4 are defined as shown in Figure 3.2(a). For the I4 × 4 prediction, the 16 samples of a 4 × 4 block are predicted using spatially adjacent samples. For each sample, eight directionally different prediction modes are supported in addition to the DC mode. In I4 × 4 prediction, a uniform prediction is performed for whole components in a block using four prediction modes. For each MB, there are 160 times of computing RD cost for intra mode decision. Thus the computational complexity is huge. It is well known that the mode decision process of intra mode is computationally complex and the number of times the RD cost is computed is five times higher than the process of inter mode [70].

Wang and Siu [71] achieved more than an 80% reduction in computing the direction for intra mode by using the characteristics of each directional prediction mode. Wu *et al.* defined homogeneous regions using the edge map and stationary regions using the sum of absolute difference of MBs. Both homogeneous and stationary characteristics are used for intra mode and early skip mode decisions. Lee and Jeon [70] reduced encoding time by 30% by adopting selective intra mode decision. Pan *et al.* [72] employed a pre-established local edge direction histogram to remove candidate modes in intra prediction.

3.5.2 Inter Mode Decision Algorithm

In the H.264/AVC video encoder, seven different block sizes can be used for inter mode in inter frame coding. Those seven different blocks are depicted in Figure 2.10. Also Figure 3.2 shows different block sizes with multiple reference frames ME concept. While larger blocks have lower motion cost, smaller blocks help to decrease the residual signal and reduce the bit rate when a MB has large motion. The mode

decision algorithm selects the best mode through an exhaustive search for all possible modes and compares the value of the cost function. To select the best inter mode, the H.264/AVC reference software uses two optimization algorithms which are: RD optimization and non-RD optimization. While RD optimization adopts the RD function in Equation 3.1 with true distortion, non-RD optimization employs SATD for the RD function in Equation 3.1.

It is obvious that the RD optimization based mode decision achieves better performance than the non-RD optimization based mode decision. When the RD optimization based mode decision is enabled, each mode performs a real encoding process, which includes ME, transformation, quantization, and entropy coding. Thus RD optimization based mode decision requires huge computational complexity. Much research has been done to reduce the complexity of RD optimization based mode decision.

Wang *et al.* achieved fast inter mode decision using successive termination and elimination. The termination of a motion search is based on either residual signal or spatial homogeneous detection. The elimination detection is determined based on the effective cost analyses. Ri *et al.* [73] determined the best inter mode for a MB by predicting the best mode from neighboring MBs in time and space and by estimating the RD cost of a MB from the MB in previous frame. Kuo and Chan [32] used motion field distribution and correlation within a MB to determine suitable inter mode. Ri *et al.* [73] achieved around 45% reduction of encoding time compared to the H.264/AVC reference software.

3.6 Summary

In this chapter, the computational complexity of the conventional video encoder is investigated and discussed. The distribution of complexity of each block of the video encoder is provided to help the reader's understanding. ME is the function block responsible for the largest proportion of the the most complexity of the video encoder. Since ME performs RDO to decide the optimal mode for each MB, the complexity of ME in H.264/AVC is significantly high compared to the other video coding standards. Depending on the number of reference frames and search window sizes, the complexity of ME can be more than 70% of the encoder complexity. The concept of multiple reference frames based ME and variable block size mode decision are described. Various ME algorithms have been proposed for reduced ME time. For example, reference selection algorithms, mode decision algorithms and search pattern algorithms are contributors to reduction of ME time. The efficient mode decision algorithm for H.264/AVC is discussed in detail.

In H.264/AVC 4×4 integer DCT is implemented to achieve a low complexity video encoder. Some fast DCT algorithms and implementation issues are discussed in this chapter. The Hadamard transform matrix, which has only integer numbers, is introduced to explain the mismatch and fast transformation.

Chapter 4

Improved Motion Estimation Time

Motion estimation is the function block which takes the most computational complexity in video encoding. Optimization of the motion estimation block is necessary to achieve a low complexity video encoder. In this chapter a proposal for a low complexity motion estimation algorithm is introduced. To reduce the complexity of motion estimation, various techniques such as reference selection algorithm, initial motion vector decision, and residue based mode decision algorithm are employed. All figures and tables in this chapter and much of the text are based on previous work by the author [74].

4.1 Introduction

In this chapter a new approach that combines a dynamic reference frame selection algorithm together with improved mode selection based on image residue (the difference of pixel value at each position between the two frames) is described. The selection of an initial reference frame is based on the information in neighboring MBs

which allows reduction of the parameter R in Equation 3.8. The use of image residue allows reduction of the parameter I in Equation 3.8. It is demonstrated that the combination of the two methods provides a significant improvement in the computation time for ME.

The rest of this chapter is organized as follows. In Section 4.2, the algorithm for dynamic reference frame selection including estimation of an initial reference frame, an initial MV and an approach for selecting early stop thresholds are described. In Section 4.3, the improved mode decision procedure is given as well as the algorithm for combining it with the procedure from Section 4.2. Simulation results using a variety of known video test sequences are provided in Section 4.4. We finalize this chapter with a summary in Section 4.5.

4.2 Motion Estimation Using Neighbor Blocks

Several reference frame selection algorithms for motion estimation have been reported in [18], [19] and [20]. Kim *et al.* [66] used MV maps to improve the speed of multiple reference frames based motion estimation. Kim *et al.*'s algorithm requires additional memory to keep MVs of previous reference frames. Chen *et al.* [75] proposed a motion estimation algorithm named *forward dominant vector selection algorithm* (FDVS). In FDVS the MVs of multiple reference frames are calculated by performing a motion search in advance. Kue and Chen [32] investigated more efficient algorithms named *variable block size activity dominant vector selection* (VADVS) and *adaptive variable block size activity dominant vector selection* (AVADVS). By finding only the MVs of the previous reference frames, VADVS and AVADVS achieved improvements in the motion estimation algorithm complexity compared to FDVS. Other reference frame

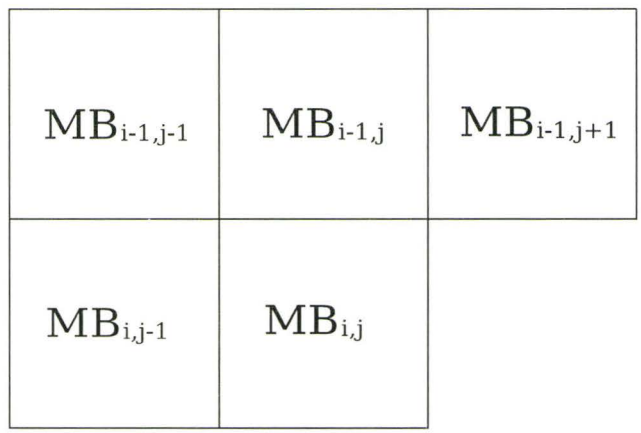


Figure 4.1: The current macroblock ($MB_{i,j}$) and its four neighborings.

selection algorithms for multiple reference frame based ME may be seen in [19], [21] and [22].

In this section a new method based on neighboring MBs for efficient ME is described. When there is high spatial correlation in a frame, there is high probability that the MV for the current MB is similar to the MVs of neighboring MBs. Figure 4.1 shows a current MB ($MB_{i,j}$) and its four neighboring MBs. The number of available neighboring MBs depends on the position of the current MB. For example, there is no neighboring MB if the current MB is located in the first column and first row ($MB_{0,0}$). In this case, the ME procedure will find the MV with conventional ME (ie. using a full search with no initial reference frame and no initial motion vector). Only one neighboring MB is available when the current MB is positioned elsewhere on the first row ($MB_{0,j}$) or the first column ($MB_{i,0}$). Otherwise there are at least two available neighboring MBs.

In the following subsections, three approaches to reduce the complexity of ME

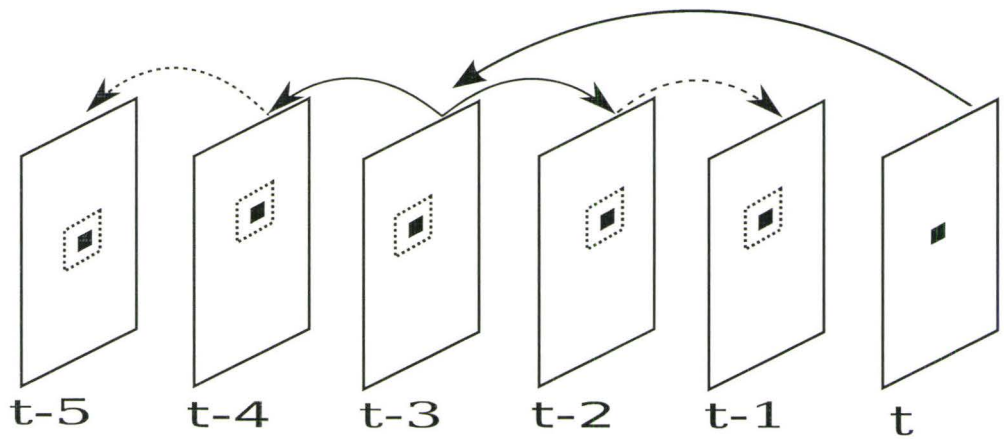


Figure 4.2: Dynamic reference frame selection. Search windows (dotted line box) around the current macroblock (solid black square) are shown for each reference frame (5 in this example). The solid arrow line indicates an initial reference selection and the dotted arrow line indicates the extension to avoid calculation of an incorrect motion vector.

that use multiple reference frames are proposed:

(1) Estimating an initial reference frame ($\Delta\hat{t}$).

(2) Estimating an initial motion vector ($\widehat{MV}$).

(3) Selecting thresholds for early stop (T_{max}, T_{min}).

4.2.1 Estimating an Initial Reference Frame

When a multiple reference frame strategy is adopted, usually more than three reference frames are used (five is typical). From a variable number of reference frames, only one frame is selected as an optimal frame for the MV of the current MB. But consideration of all reference frames is a time consuming task since computational

Algorithm 1 Dynamic reference frame selection for each macroblock

Step 1 Find initial reference frame ($\Delta\hat{t}$) using neighboring macroblocks
Step 2 Calculate the Lagrangian cost functions $j_{\Delta\hat{t}-1\geq 1}, j_{\Delta\hat{t}}$ and $j_{\Delta\hat{t}+1\leq R}$ for the macroblock:
if $j_{\Delta\hat{t}} \leq j_{\Delta\hat{t}-1}$ and $j_{\Delta\hat{t}} \leq j_{\Delta\hat{t}+1}$ **then**
  $\Delta t \leftarrow \Delta\hat{t}$
else if $j_{\Delta\hat{t}} \geq j_{\Delta\hat{t}+1}$ **then**
  repeat
    $\Delta\hat{t} \leftarrow \Delta\hat{t} + 1$
  until $j_{\Delta\hat{t}} \leq j_{\Delta\hat{t}+1}$
else if $j_{\Delta\hat{t}} \geq j_{\Delta\hat{t}-1}$ **then**
  repeat
    $\Delta\hat{t} \leftarrow \Delta\hat{t} - 1$
  until $j_{\Delta\hat{t}} \leq j_{\Delta\hat{t}-1}$
  $\Delta t \leftarrow \Delta\hat{t}$
end if

complexity of ME increases with the number of reference frames. Thus, by selecting an initial frame before starting ME, the computational complexity can be reduced significantly.

The neighboring MBs, which already have associated reference frames, can provide information to estimate the initial reference frame for the current MB. In the proposed method, the initial reference frame number is selected as

$$\Delta\hat{t}^* = \left\lfloor \frac{1}{B} \sum_{i=1}^{B} \Delta t_i^* \right\rfloor \tag{4.1}$$

where B is the number of neighboring blocks and Δt_i^* is the reference frame number of the ith block. Defining the value of the Lagrangian cost function $j_{\Delta\hat{t}^*}$ in frame $\Delta\hat{t}^*$ as

$$j_{\Delta\hat{t}^*} = J(MB, I|\lambda_{mode}, QP, \Delta\hat{t}^*) \tag{4.2}$$

then $j_{\Delta\hat{t}^*}$ is the cost function of macroblock MB in the frame $\Delta\hat{t}^*$ and J is defined as in Equation 3.1.

There is a high probability of finding the incorrect MV if the wrong reference frame is selected. Thus, extending the motion search to include one additional previous and future frame is advantageous. Figure 4.2 shows the extended reference frame selection (indicated by the dotted arrows) after finding the MV in the estimated initial reference frame, to avoid this situation. The cost functions from each reference frame are then compared and the MV which has the smallest cost function is selected. Algorithm 1 summarizes the procedure for dynamic reference frame selection for each MB.

4.2.2 Estimating an Initial Motion Vector

After moving to the initial search point, in the ME algorithm, all candidate points in the search window are checked and compared. Thus incorrect initial search point selection will require more time to find the best MV for the current block. In the proposed ME algorithm, with the assumption that the MV of the current MB is similar to those of neighboring MBs, the initial MV of the current MB $(\widehat{mv_x}^*, \widehat{mv_y}^*)$ is estimated as

$$\begin{aligned} \widehat{mv_x}^* &= \mathcal{M}(mv_x^*) \\ \widehat{mv_y}^* &= \mathcal{M}(mv_y^*) \end{aligned} \tag{4.3}$$

where the operator $\mathcal{M}()$ indicates the mean value and mv_x^* and mv_y^* are the chosen MVs of neighboring MBs in the horizontal and vertical directions, respectively. The estimated MV $\widehat{mv_x}^*$ and $\widehat{mv_y}^*$ are used as the initial search point for ME of the current MB.

Algorithm 2 Set early stop conditions for each macroblock

Step 1 Calculate initial MV using MVs of neighboring macroblocks.
Step 2 Calculate the MAE on the reference frame using defined search pattern.
Step 3 Set MAE^* as the smallest MAE of search points.
if the MAE^* exists at the centre point **then**
(1) Set centre point as mv_x and mv_y.

(2) Stop ME.
else if $MAE^* \leq T_{min}$ **then**
(1) Set the point having MAE^* as mv_x and mv_y.

(2) Stop ME.
else if $T_{min} \leq MAE^* \leq T_{max}$ **then**
(1) Set the point having MAE^* as the centre point of new ME.

(2) Change search pattern from diagonal search pattern to hexagonal search pattern.

(3) Goto step 3.
else if $MAE^* \geq T_{max}$ **then**
(1) Set the point having MAE^* as the centre point of new ME.

(2) Change search pattern from diagonal search pattern to hexagonal search pattern.

(3) Increase the size of hexagon.

(4) Goto Step 3.
end if

4.2.3 Selecting Thresholds for Early Stop

Most fast ME algorithms do not consider all search points on the search pattern. By introducing an early stop condition, the speed of ME can be improved. With the assumption that ME error increases monotonically, there is no need to process additional search points after finding an acceptable MV. Detection of this early stop condition is based on the information in neighboring MBs indicating if the error of the current MB is acceptable.

For block (B) having MV (mv_x^*, mv_y^*) and block size $(N_h \times N_v) \in \{\mathsf{P16} \times \mathsf{16}, \mathsf{P16} \times \mathsf{8}, \mathsf{P8} \times \mathsf{16}, \mathsf{P8} \times \mathsf{8}, \mathsf{P8} \times \mathsf{4}, \mathsf{P4} \times \mathsf{8}, \mathsf{P4} \times \mathsf{4}\}$, the SAD is calculated as

$$\begin{aligned} &SAD_B^*(mv_x^*, mv_y^*) = \\ &\textstyle\sum_{i=1}^{N_v} \sum_{j=1}^{N_h} \left| f_t(i,j) - \hat{f}_{t-\Delta t^*}(i + mv_x^*, j + mv_y^*) \right| \end{aligned} \tag{4.4}$$

where $\hat{f}_{t-\Delta t^*}$ is the reconstructed frame with reference number Δt^*. Two thresholds are defined for the stop condition of the search as

$$\begin{aligned} T_{min} &= \min_{B \in MB_n} SAD_B^*(mv_x^*, mv_y^*) \\ T_{max} &= \max_{B \in MB_n} SAD_B^*(mv_x^*, mv_y^*) \end{aligned} \tag{4.5}$$

where MB_n is the neighboring MB. The number of MB depends on the position of the current MB.

Algorithm 2 summarizes the ME procedure using early stop conditions. As mentioned earlier, all information for early stop conditions are from the neighboring MBs. Some experimental results of the performance of our ME algorithm using neighboring MBs was reported in [44].

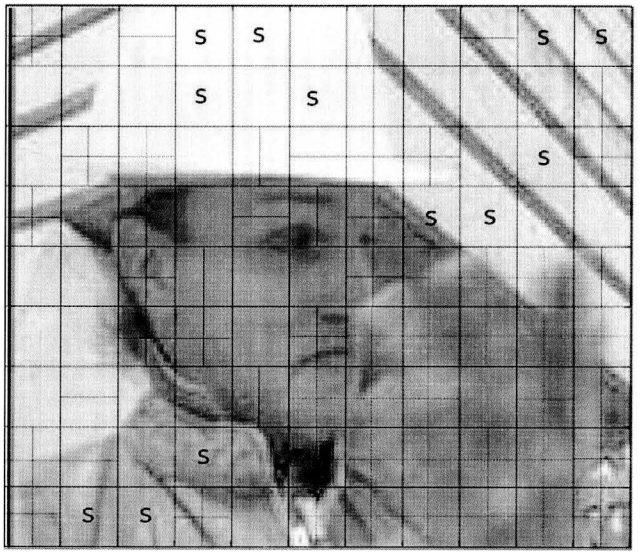
4.3 Residue Based Mode Decision

4.3.1 Analysis of Mode Decision

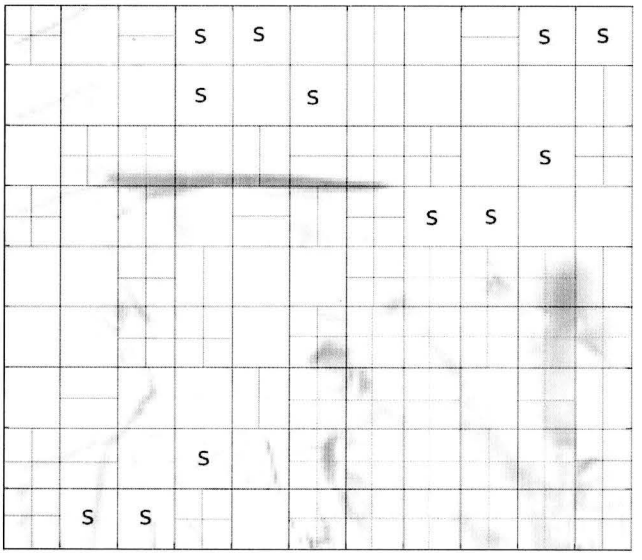
Mode decision is the part of ME that uses variable block size. In real video sequences, the distribution of MB modes depends on the characteristics of the input video sequence. Figure 4.3 shows an example of different modes which are superimposed on the original 256th frame of the **Foreman** sequence and the residue image between the reconstructed 255th and the original 256th frames. The MB mode for each MB is determined by the H.264/AVC reference software JM13.1 [76]. In Figure 4.3, the skip mode MB is represented by the letter S in the block. From Figure 4.3, it can be seen that the best mode is more related to the residue of a MB than the characteristics of the original MB detail such as edges.

Note that although there are edges on the right side of the helmet shown in the frame, the MB is encoded with P16 $\times$ 16 mode instead of a sub-block mode. The reason is that the bit rate, which is $R()$ in Equation 3.1, is related to the coefficients of the Hadamard or DCT transform of the residue of the MB. Also sub-block modes are usually adopted for MB areas which exhibit motion.

Usually SKIP and P16 $\times$ 16 modes are employed when the MV is similar to those of predicted MVs and the residue between original and predicted MB is small. The smaller block size modes such as P8 $\times$ 8 are suitable for MBs that contain complicated and different objects. After checking inter modes, intra mode can be used for a MB if the cost function of the current MB is too large, such as in the case when there is low temporal redundancy. The best mode depends on how the ME procedure estimates the MVs and reduces the difference between the current MB and estimated MB with



(a)



(b)

Figure 4.3: Example of different modes: (a) original and (b) residue image.

Table 4.1: The distribution (in percentage) of modes selected for several sequences with various characteristics (QP=26)

Sequence		SKIP	P16 × 16	P16 × 8	P8 × 16	P8 × 8	INTRA
QCIF	**Carphone**	23.6	29.0	11.1	12.5	22.4	1.3
	Container	73.4	11.3	4.5	3.9	6.5	0.4
	Grandma	73.7	10.3	3.8	4.5	7.7	0.0
	M&D	56.6	14.8	7.6	7.4	13.5	0.0
	Salesman	72.5	8.4	2.9	3.0	13.2	0.0
	Silent	58.9	12.8	5.0	6.7	15.6	0.9
CIF	**Flower**	26.4	22.4	9.5	5.4	35.9	0.3
	Football	19.3	26.0	7.6	11.9	11.7	23.5
	Hall	36.8	36.1	10.7	5.7	8.8	2.0
	Highway	48.5	24.7	10.6	5.1	7.5	3.6
	Mobile	5.8	34.1	12.8	12.2	35.0	0.0
	News	73.3	9.7	3.9	5.1	7.5	0.5

the given MB modes as well as QPs.

Table 4.1 and 4.2 show the distribution of MB modes for the video test sequences with varying QPs. The MB mode distribution is obtained when the encoder is configured as follows: (1) each sequence is encoded using the number of maximum frames given in Table A.1, (2) the QPs are 26 and 34, (3) the number of inter frames between successive intra frames is 14, (4) the search window size is ±32, (5) all sub-block modes are turned on, and (6) the number of reference frames is 5.

In Table 4.1 and 4.2, the P8 × 8 mode includes all sub-block modes, which are P8 × 8, P8 × 4, P4 × 8 and P4 × 4 modes. In both Tables, it can be seen that most MBs are encoded with either SKIP or P16 × 16 modes in most of the sequences. There are more MBs coded with SKIP mode when the QP is increased. Also the sub-block mode (P8 × 8) is used less than 10% of the modes in most of the sequences except for the **Mobile** and **Flower** sequences, which have large and complicated motion (these

Table 4.2: The distribution (in percentage) of modes selected for several sequences with various characteristics (QP=34)

Sequence		SKIP	P16 × 16	P16 × 8	P8 × 16	P8 × 8	INTRA
QCIF	**Carphone**	43.8	33.8	7.4	7.9	5.8	1.3
	Container	87.8	7.0	2.1	1.5	1.4	0.2
	Grandma	86.7	9.0	1.9	2.0	0.5	0.0
	M&D	74.8	16.6	3.6	3.6	1.4	0.0
	Salesman	82.0	8.7	2.9	3.0	3.3	0.0
	Silent	70.6	13.8	4.5	5.6	4.3	1.2
CIF	**Flower**	36.5	21.3	8.5	5.3	28.0	0.3
	Football	35.5	23.0	6.2	8.0	5.7	21.5
	Hall	83.7	8.7	2.3	1.7	3.3	0.4
	Highway	72.7	17.2	4.9	2.4	1.3	1.6
	Mobile	20.2	36.2	12.4	12.9	18.3	0.0
	News	82.5	8.7	2.3	3.1	2.7	0.7

have 28.0% and 18.3% of sub-block mode, respectively with large QP). The sub-block mode is increased when sequences are coded with small QPs. Also intra mode takes less than 1.5% except for the **Football** sequence, which contains large and complicated motion. Thus calculating RD cost for sub-block and intra prediction is wasteful of computational resources in most cases. In particular, skip mode is sufficient when the frame contains small motion such as in the **Container** sequence (see Table 4.2).

The ME searches for the best MV for each mode and stores the required bit rate to encode the residue. While increasing the number of blocks reduces the residue of the current MB, it increases the number of MVs. Thus the bit rate and the best mode of the current MB are related to the residue more so than with the image detail of the MB itself. In the following sections, the proposed mode decision algorithm based on residue is described in detail.

4.3.2 Inter Mode Decision Based on Residue Image

Each MB is coded with different modes such as intra mode, skip mode, block mode and sub-block mode. When there is a large difference between the current MB and the predicted MB from the reference frame, the MB is coded with intra mode. The inter mode decision is a complicated process since it requires the comparison of the cost function of each mode. A residue image based inter mode decision algorithm was proposed in [45] to enhance fast inter mode decision. In the proposed fast mode decision algorithm, the residue information is used to check whether the current MB is suitable for skip mode or not.

When skip mode is selected for the current MB, the encoder saves ME time by skipping the rate-distortion comparison for other modes. If skip mode is not selected for the current MB, the proposed algorithm then decides the best mode between block and sub-block mode by investigating the distribution of the residue. In the following subsection, the skip mode decision and mode selection strategies between block and sub-block modes are explained in detail.

Skip mode decision

Skip mode is selected when the current MB is the same as the predicted MB. In the H.264/AVC recommendation, there are four conditions to encode the current MB as skip mode. The conditions are:

(1) the best mode is P16 $\times$ 16,

(2) the MV should be the same as the predicted MV,

(3) the selected reference frame immediately precedes the current frame,

(4) the transform coefficients of the blocks are all zero.

For skip mode detection, the encoder predicts the MV of the current MB using the MVs from the available neighboring MBs, which are shown in Figure 4.1. With the predicted MV and reconstructed previous frame, the residue between the current MB and the predicted MB is found from

$$E(x,y) = |(f_t(x,y) - \hat{f}_{t-1}(x - pmv_x, y - pmv_y)| \tag{4.6}$$

where f_t is the frame at time t and $\hat{f}_{t-1}$ is the reconstructed previous frame, x and y are the position of the MB in the horizontal and vertical directions, respectively. The pmv_x and pmv_y are predicted MVs in the horizontal and vertical directions, respectively.

There are sixteen 4×4 blocks E_i (for $i = 0, \cdots, 15$) in each MB since the MB size is 16×16. Then the Hadamard transform for each block E_i is

$$C_i(u,v) = HE_i(x,y)H^{\mathrm{T}} \tag{4.7}$$

where H is the Hadamard matrix which is

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}. \tag{4.8}$$

The quantized coefficients of each transformed block C_i can be found from

$$Q_i(u,v) = \frac{C_i(u,v) \times Ls(u,v) - Lo(u,v)}{2^{qbits}} \tag{4.9}$$

where Ls and Lo are a predefined level scale and a level offset, respectively, and $qbits$ is a constant ($\frac{QP}{6} + 15$, as set in the JM13.1 reference software [76]).

Thus if the transformed coefficient $C_i(u,v)$ satisfies the condition

$$C_i(u,v) < \frac{2^{qbits} + Lo(u,v)}{Ls(u,v)} \tag{4.10}$$

the quantized coefficient $Q_i(u,v)$ is zero. Now the sum of the residues of each block is

$$d_i = \sum_{x=0}^{3} \sum_{y=0}^{3} E_i(x,y), \tag{4.11}$$

where $E_i, 0 \leq i \leq 15$ are the blocks of the MB residue E. The sum of the residue d_i is the DC component of the Hadamard transform. Finally, the threshold for skip mode detection T_{intra} is

$$T_{intra} = \frac{2^{qbits} + Lo(0,0)}{Ls(0,0)}. \tag{4.12}$$

Since d_i is the largest value among the quantized coefficients, it is clear that if all elements E_i are less than T_{intra}, the quantized coefficients of C_i are all zero. Thus the skip mode is selected for the current MB when all d_i in the current MB are less then T_{intra}.

Algorithm 3 Find the best mode for each macroblock

Step 1 Find skip motion vector.
Step 2 Get luminance residue E for macroblock.
Step 3 Calculate block SAE for each 4×4 block (d_i).
if $d_i \leq T_{intra}$ for all sixteen blocks **then**
  Set current macroblock mode to SKIP.
else if $\sum d_i \geq (3 \times T_{intra} \times QP)$ **then**
  Set current macroblock mode to P8 $\times$ 8.
else if Each 8×8 block has more than three 4×4 blocks satisfying $d_i \geq T_{intra}$ **then**
  Set current macroblock mode to P16 $\times$ 16.
else
  Do mode decision for all inter modes.
end if
Step 4 Select the best mode by comparing to intra modes.

Algorithm 4 Proposed fast motion estimation

for all $MB_{i,j}$ **do**
  if inter MB **then**
    (1) Estimate the initial reference frame using Algorithm1.
    (2) Set thresholds for early stop conditions using Algorithm2.
    (3) Find skip motion vector.
    (4) Calculate residue of the current MB.
    (5) Find best modes for the current MB using Algorithm3.
    if the best mode is block mode **then**
      Calculate the cost function J for block mode.
    else
      Calculate the cost function J for sub-block mode.
    end if
  else
    Encode the current macroblock using intra mode.
  end if
  Choose the best macroblock mode for the current macroblock.
end for

Mode decision between block and sub-block mode

After the skip mode decision, the proposed mode decision algorithm proceeds to decide whether the current MB is encoded as either block mode or sub-block mode (if the current MB is determined not to be skip mode). The block mode is preferred to sub-block mode when the residue is small. Calculation of the cost function of sub-block mode takes more time than the calculation of the cost function of a block mode. In the proposed algorithm, sub-block mode is adopted when the MB contains substantial residual error.

Thus the sub-block mode is distinguished from the block mode by considering the sum of the residue for each 4×4 block and its distribution. The sum of absolute error for a MB is defined as $SAE = \sum_i d_i$. Then sub-block and block modes are selected when the conditions

(1) $SAE \geq 3 \times T_{intra} \times QP$ or

(2) Each 8×8 block has more than three 4×4 blocks with $d_i > T_{intra}$

are satisfied. In the first condition, QP represents the quantization parameter and the constant 3 was selected empirically. Finally, the MB is assumed as one of P16 $\times$ 8 or P8 $\times$ 16 mode when the residue of the MB does not satisfy both skip and sub-block mode conditions. Algorithm 3 specifies the proposed mode decision method. The performance of the residue based mode decision algorithm can be found in [45].

A new ME algorithm is proposed by combining the neighboring block information based fast ME algorithm with residue based mode decision algorithm to minimize the complexity of the encoder. Algorithm 4 summarizes the whole procedure of the proposed fast ME algorithm which combines initial reference frame selection, initial

Table 4.3: Hardware specification and encoder condition

Hardware specification	
CPU	Intel Core 2Duo T7400 2.16 GHz
Memory	2 GB
OS	Mac OS X
Compiler	GCC
Encoder configuration	
Profile @ Level	Main @ 2.0
Sequence type	IBBPBBP ...
Intra period	30
Frame rate	30 frame per second (fps)
Quantization parameter	$20 \sim 40$ (+2 steps)
Motion estimation	fast full, UMHexagon, EPZS
Search range	± 8, ± 16, ± 32
Number of reference frames	2, 5
Entropy coding mode	CABAC

MV estimation, the detection of the early stop condition and the mode decision.

4.4 Simulation Results and Discussion

To verify performance, the proposed ME algorithm was implemented in the H.264/AVC reference encoder of JM13.1 [76]. Twelve sequences of various formats and characteristics were tested to compare the performance of the proposed algorithm with that of the reference software. The number of encoded frames and characteristics of each sequence are summarized in Table A.1. The test sequences in Table A.1 were chosen to cover a variety of different background and object motions. The set comprises six each of QCIF and CIF formats.

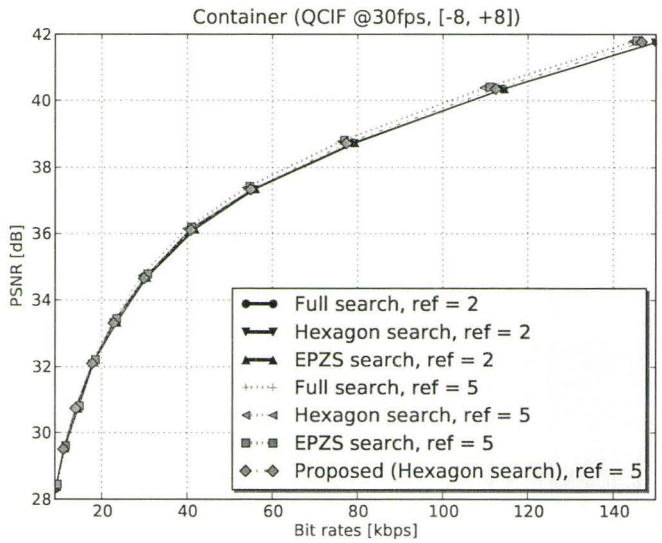
Table 4.3 shows the experimental hardware specification and encoder configuration that was used. All sequences listed in Table A.1 were encoded with IBBPBBP structure and main profile @ level 2. The frame rate was 30 frames per second, and the search ranges for ME were $\pm 8, \pm 16$ and ± 32 pixels. RDO was turned on and rate control was disabled. Simulations were run using both two and five reference frames. Note that the minimum number of reference frames for B frame coding is two. The range of QP was from 20 to 40. These QPs were chosen to generate appropriate bit rates for realistic applications such as video telephony (normally 25 kbps) and internet video (normally 3 Mbps). Although all test sequences were evaluated for each of the search ranges, Figures 4.4 - 4.9 provide several representative measurement results, chosen for brevity, as examples for presentation here.

To evaluate coding efficiency, the performance was measured based on distortion and bit rates. In Figures 4.4 - 4.9, the distortion is represented using PSNR of the Y (luminance) component. Figures 4.4 - 4.6 show the RD curves of the proposed ME and reference software when the format of the input sequence is QCIF and the search ranges are $\pm 8, \pm 16$ and ± 32 pixels, for several examples of the sequences. Figure 4.4 (a) shows the RD curves for the test sequences **Container** with ± 8 pixels search range. The hexagon search pattern is adopted in the proposed motion estimation algorithm. Figure 4.4 (b), and Figures 4.5 (a) and (b) are RD curves when the search range is ± 16 pixels for the test sequences **Carphone**, **Foreman** and **Mother&daughter**, respectively. Examples of RD curves with ± 32 pixels search range are shown in Figures 4.6 (a) and (b). The PSNR difference between JM13.1 searching five reference frames and the new proposed algorithm is hardly distinguishable at the low bit rates while there is less than 0.4 dB in all cases at the high bit

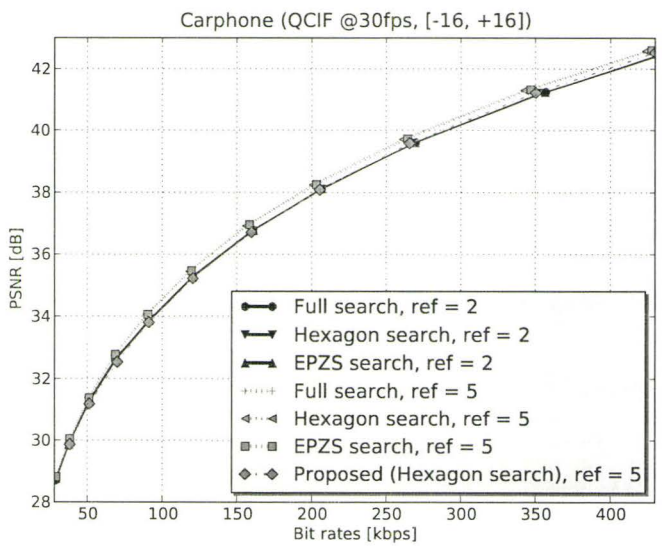
rates. The proposed algorithm exhibits better performance than JM13.1 searching two reference frames for all bit rates.

Similarly, Figures 4.7 - 4.9 show RD curves for several examples of CIF test sequences. The PSNR of JM13.1 searching five reference frames and the proposed algorithm is hardly distinguishable for the **Flower**, **Highway** and **News** sequences. There is a small degradation of PSNR for the **Hall** and **Mobile** sequences. The PSNR of the proposed algorithm is better than the PSNR of JM13.1 searching two reference frames for all sequences. In particular, the performance of the proposed motion estimation algorithm is comparable to the performance of reference software searching five reference frames and much better than the performance of reference software searching two reference frames, as shown in Figures 4.7 - 4.9. For the **Football** sequence, the proposed algorithm provides slightly reduced performance compared to the JM13.1. In the **Football** sequence there are many objects with large motion vectors. Usually MBs with large motion vectors are encoded with intra mode. Also the correlation between neighboring MBs would be expected to be small. Thus, the information from neighboring MBs can be inaccurate for the current MB.

Generally, the proposed algorithm performs better for input sequences that have small object motion on a slowly varying background. As expected, the search range does not significantly effect the coding efficiency. Tables 4.4 $\sim$ 4.7 summarize the average *encoding time* (ET) and average *motion estimation time* (MET) of the proposed algorithm and for H.264/AVC reference software using various search patterns. The average ET and MET are obtained when the reference software and proposed algorithm based encoders are executed on an Intel Core2Duo T7400 2.16 GHz with 2 GB main memory (see Table 4.3.) The speed gains SG_{ET} and SG_{MET} on Table 4.6

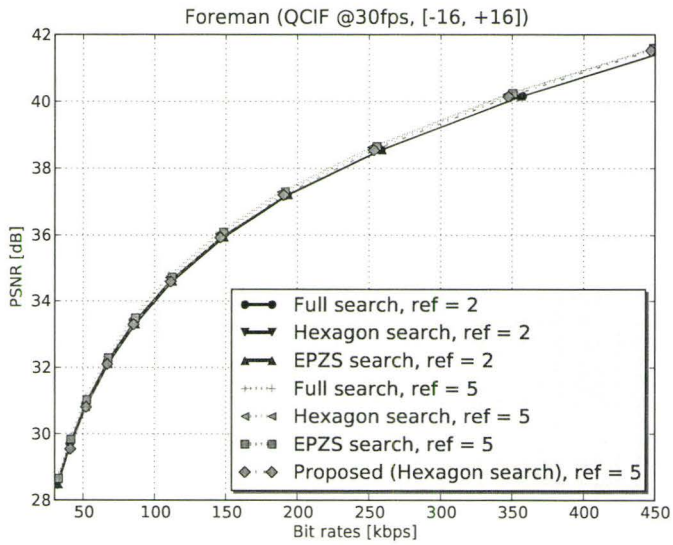


(a)

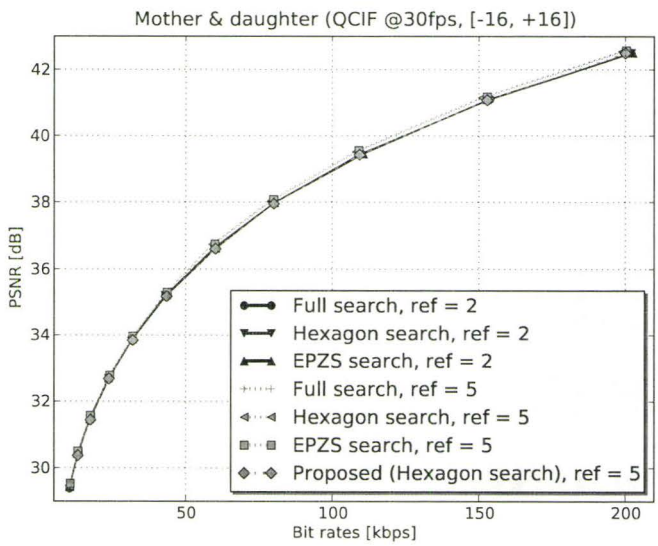


(b)

Figure 4.4: Comparison of the coding efficiency of the proposed algorithm with H.264/AVC using examples of QCIF sequences with various search ranges: (a) **Container** and (b) **Carphone**.

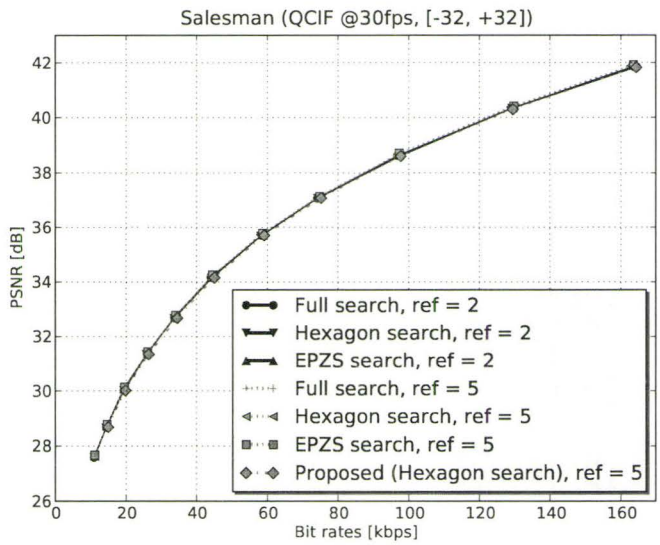


(a)

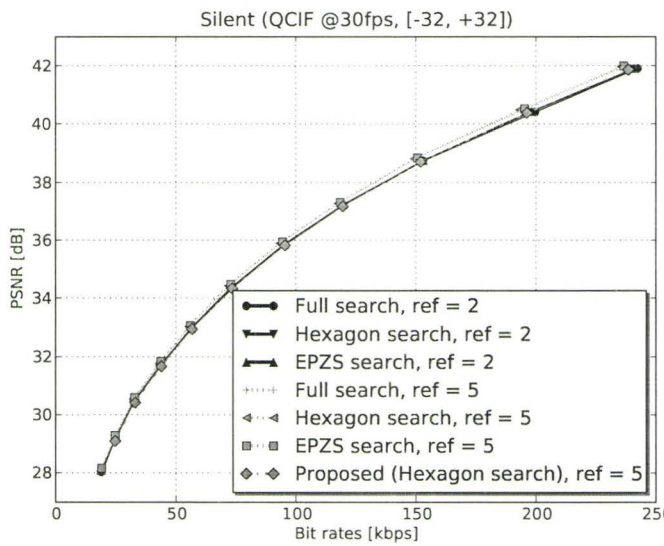


(b)

Figure 4.5: Comparison of the coding efficiency of the proposed algorithm with H.264/AVC using examples of QCIF sequences with various search ranges: (a) **Foreman** and (b) **Mother&daughter**.

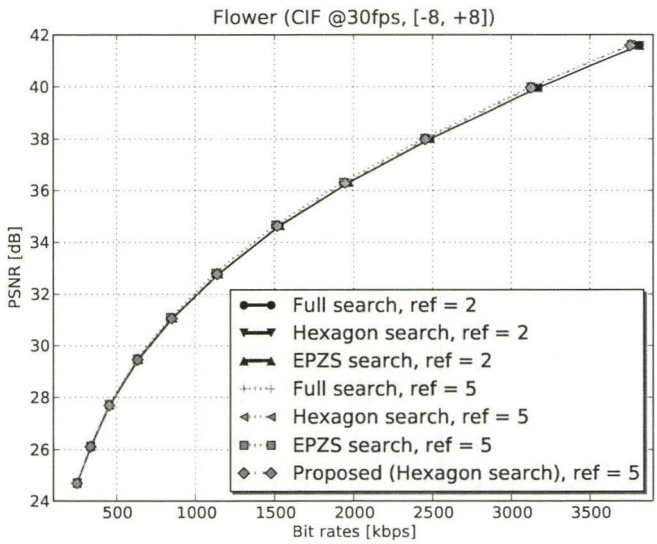


(a)

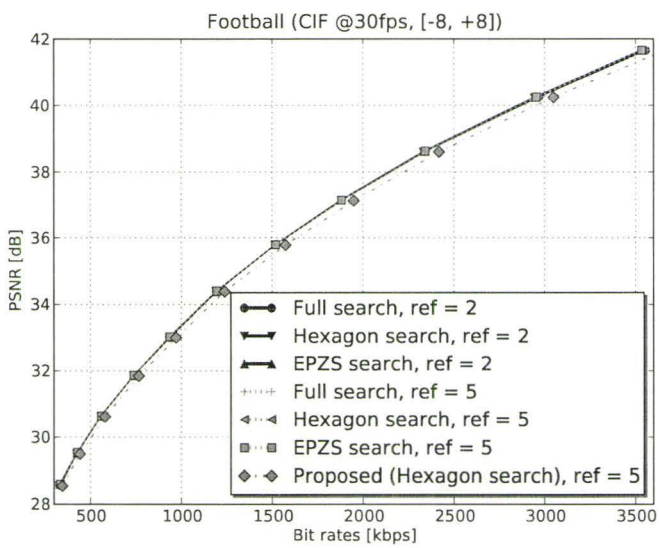


(b)

Figure 4.6: Comparison of the coding efficiency of the proposed algorithm with H.264/AVC using examples of QCIF sequences with various search ranges: (a) **Salesman** and (b) **Silent**.

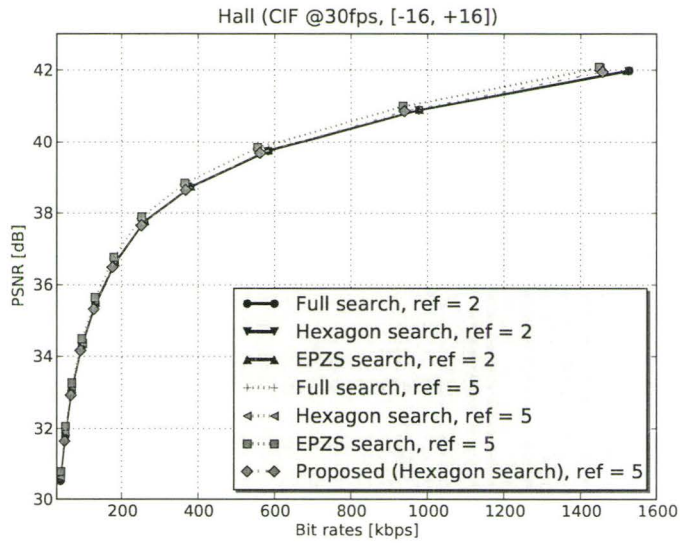


(a)

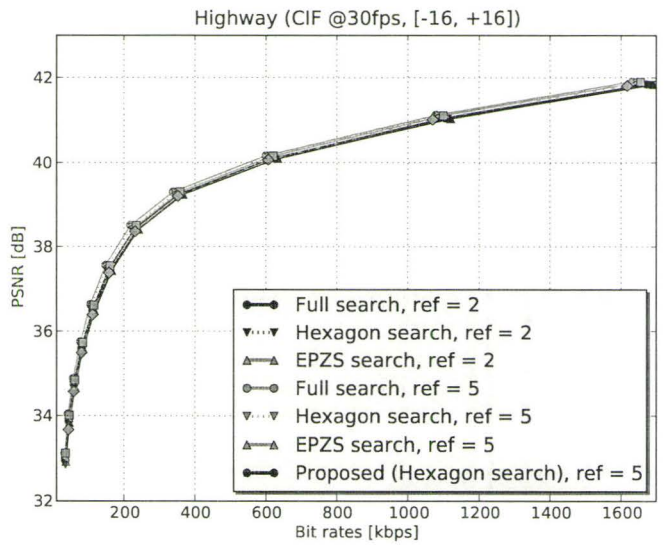


(b)

Figure 4.7: Comparison of the coding efficiency of the proposed algorithm with H.264/AVC using examples of CIF sequences with various search ranges: (a) **Flower** and (b) **Football**.

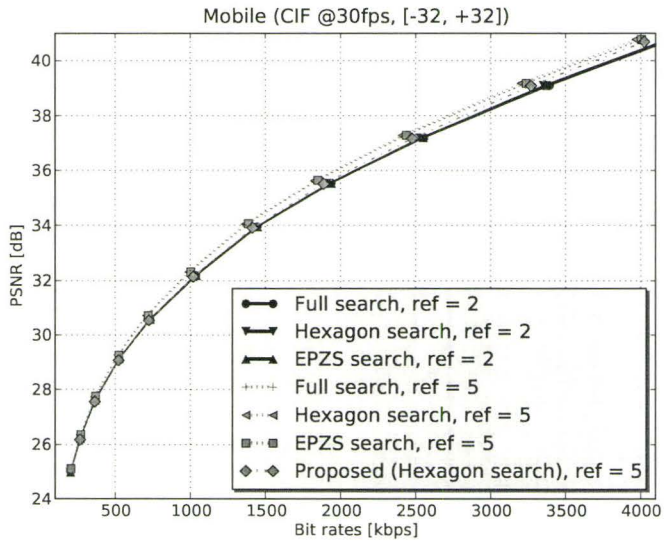


(a)

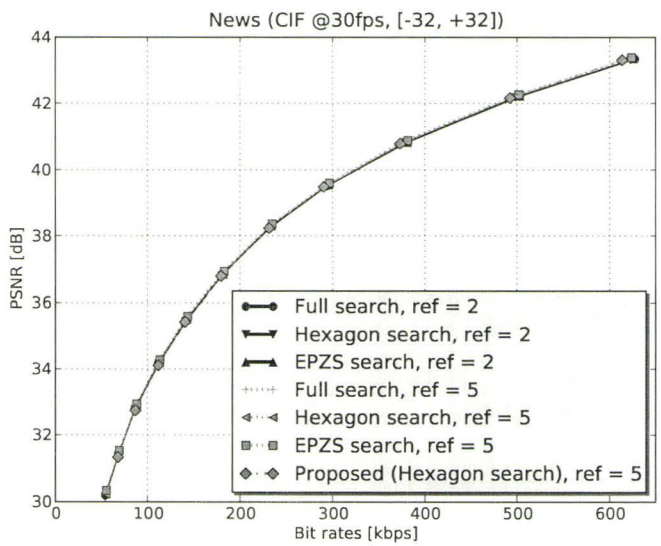


(b)

Figure 4.8: Comparison of the coding efficiency of the proposed algorithm with H.264/AVC using examples of CIF sequences with various search ranges: (a) **Hall** and (b) **Highway**.



(a)



(b)

Figure 4.9: Comparison of the coding efficiency of the proposed algorithm with H.264/AVC using examples of CIF sequences with various search ranges: (a) **Mobile** and (b) **News**.

and 4.7 are obtained as

$$\begin{aligned} \mathrm{SG_{ET}} &= \frac{\mathrm{ET_{H.264.Hex}} - \mathrm{ET_{Proposed.Hex}}}{\mathrm{ET_{H.264.Hex}}}, \\ \mathrm{SG_{MET}} &= \frac{\mathrm{MET_{H.264.Hex}} - \mathrm{MET_{Proposed.Hex}}}{\mathrm{MET_{H.264.Hex}}} \end{aligned} \tag{4.13}$$

where $\mathrm{ET_{H.264.Hex}}$ and $\mathrm{MET_{H.264.Hex}}$ are ET and MET of H.264 reference software with Hexagon search, respectively, and $\mathrm{ET_{Proposed.Hex}}$ and $\mathrm{MET_{Proposed.Hex}}$ are ET and MET of proposed algorithm with Hexagon search, respectively.

The complexity of different motion estimation algorithms are compared while searching with two and five reference frames and the QP is 30. Tables 4.4 and 4.5 show the comparison of the ET and MET when two reference frames are used for the proposed ME. When the search range was ± 16 the mean MET was reduced by approximately 25% to 33% (Table 4.4). When the search range was extended to ± 32 (Table 4.5), the proposed algorithm reduces the MET to approximately 33% to 35% of that of the reference software.

Five frames are common as the reference frame for multiple reference frame based ME. We also choose five as illustrative example. In case of ME with five reference frames the speed gain was increased. When the search range was ± 16 and the search pattern of both the reference H.264 and the proposed algorithm are the same, the proposed algorithm reduces the mean MET to approximately 64% to 73% compared to that of the JM13.1 reference software (Table 4.6). When the search range was extended to ± 32 (Table 4.7), the proposed algorithm reduces the MET to approximately 60% to 74% of that of the reference software. We anticipate better performance when more than five frames are used for reference frame.

Using a full search and EPZS search pattern, the proposed motion estimation

Table 4.4: Comparison of the average motion estimation time (MET) and average encoding time (ET) with search range of ±16 and number of reference frames = 2 (all times given in milliseconds)

Sequence	H.264/AVC						Proposed algorithm	
	Full search		EPZS		Hexagon search		Hexagon search	
	ET	MET	ET	MET	ET	MET	ET (SG_{ET})	MET (SG_{MET})
Carphone	126.60	116.62	46.37	36.86	43.81	34.38	34.32 (21.7%)	25.12 (26.9%)
Container	127.14	116.50	39.12	29.48	37.32	28.00	30.38 (18.6%)	20.73 (26.0%)
Foreman	149.72	139.77	48.80	39.18	46.35	36.83	35.93 (22.5%)	26.39 (28.3%)
M&D	152.26	141.26	39.42	29.84	38.07	28.80	30.88 (18.9%)	21.51 (25.3%)
Salesman	148.06	137.87	39.86	30.25	37.53	27.89	30.54 (18.6%)	20.99 (24.7%)
Silent	148.63	137.66	42.74	33.02	41.30	31.64	33.06 (20.0%)	23.49 (25.8%)
Flower	424.37	389.73	156.32	121.70	158.64	124.55	114.76 (27.7%)	82.28 (33.9%)
Football	485.14	449.88	213.15	177.43	227.87	192.11	157.14 (31.0%)	123.93 (35.5%)
Hall	642.58	602.56	161.53	121.38	153.28	112.94	114.06 (25.6%)	76.93 (31.9%)
Highway	698.82	658.03	180.91	140.93	169.34	129.21	123.57 (27.0%)	86.54 (33.0%)
Mobile	626.77	581.70	215.16	171.10	217.41	173.21	152.65 (29.8%)	112.18 (35.2%)
News	657.41	616.95	168.92	128.45	158.47	118.09	115.11 (27.4%)	78.00 (33.9%)

Table 4.5: Comparison of the average motion estimation time (MET) and average encoding time (ET) with search range of ±32 and number of reference frames = 2 (all times given in milliseconds)

Sequence	H.264/AVC						Proposed algorithm	
	Full search		EPZS		Hexagon search		Hexagon search	
	ET	MET	ET	MET	ET	MET	ET (SG_{ET})	MET (SG_{MET})
Carphone	372.04	360.74	51.12	40.92	52.67	42.56	37.43 (28.9%)	28.01 (34.2%)
Container	372.84	362.15	42.81	32.77	43.90	33.90	32.28 (26.5%)	22.97 (32.2%)
Foreman	401.85	390.44	54.59	43.93	57.22	46.57	40.00 (30.1%)	29.85 (35.9%)
M&D	407.75	396.98	42.98	33.36	46.11	35.50	32.60 (29.3%)	23.30 (34.4%)
Salesman	400.54	389.43	43.55	33.95	45.25	34.40	31.99 (29.3%)	22.63 (34.2%)
Silent	399.14	387.50	46.26	36.16	49.40	38.55	35.05 (29.0%)	25.60 (33.6%)
Flower	1250.84	1214.04	167.13	131.45	195.11	159.37	130.12 (33.3%)	97.54 (38.8%)
Football	1345.75	1307.02	224.20	187.89	269.38	232.28	176.29 (34.6%)	142.89 (38.5%)
Hall	1606.50	1564.76	165.37	124.46	163.35	122.11	119.74 (26.7%)	82.31 (32.6%)
Highway	1676.70	1634.96	184.66	144.41	180.76	140.13	129.73 (28.2%)	92.84 (33.7%)
Mobile	1564.64	1518.21	220.77	176.74	255.48	211.20	176.54 (30.9%)	135.86 (35.7%)
News	1614.37	1573.24	171.15	130.72	166.30	126.03	120.65 (27.5%)	83.58 (33.7%)

Table 4.6: Comparison of the average motion estimation time (MET) and average encoding time (ET) with search range of ±16 and number of reference frames is 5 (all times given in milliseconds)

Sequence	H.264/AVC						Proposed algorithm	
	Full search		EPZS		Hexagon search		Hexagon search	
	ET	MET	ET	MET	ET	MET	ET (SG_{ET})	MET (SG_{MET})
Carphone	340.19	325.51	103.79	89.81	102.50	89.21	40.18 (60.8%)	30.55 (65.8%)
Container	340.73	327.10	79.26	66.74	78.02	64.88	32.80 (58.0%)	23.44 (63.9%)
Foreman	342.90	328.60	115.16	98.57	115.11	101.97	43.28 (62.4%)	33.60 (67.0%)
M&D	333.93	319.79	86.98	73.16	87.15	73.93	34.33 (60.6%)	24.77 (66.5%)
Salesman	332.12	317.72	83.27	70.34	83.10	69.94	33.97 (59.1%)	24.49 (65.0%)
Silent	335.09	321.17	91.63	78.35	93.77	80.21	36.67 (60.9%)	27.13 (66.2%)
Flower	944.46	905.83	283.38	245.90	331.66	293.90	132.94 (59.9%)	99.93 (66.0%)
Football	1113.17	1073.08	382.54	342.58	457.17	417.58	182.96 (60.0%)	148.72 (64.4%)
Hall	1374.07	1320.30	309.28	257.75	297.06	245.84	119.33 (59.8%)	81.71 (66.8%)
Highway	1451.98	1399.18	354.87	302.76	329.98	278.53	127.33 (61.4%)	90.06 (67.7%)
Mobile	1362.45	1306.25	460.14	404.03	500.19	444.13	172.94 (65.4%)	133.61 (69.9%)
News	1383.29	1329.84	338.43	286.27	320.94	269.58	121.09 (62.3%)	83.65 (69.0%)

Table 4.7: Comparison of the average motion estimation time (MET) and average encoding time (ET) with search range of ±32 and number of reference frames is 5 (all times given in milliseconds)

Sequence	H.264/AVC						Proposed algorithm	
	Full search		EPZS		Hexagon search		Hexagon search	
	ET	MET	ET	MET	ET	MET	ET (SG_{ET})	MET (SG_{MET})
Carphone	1032.56	1012.95	107.43	93.45	120.94	107.28	45.32 (62.5%)	35.65 (66.8%)
Container	1091.39	1071.43	83.35	69.60	85.15	71.64	39.12 (54.1%)	28.80 (59.8%)
Foreman	1103.31	1082.47	116.68	102.48	141.77	127.71	55.84 (60.6%)	45.07 (64.7%)
M&D	1083.06	1062.41	90.04	75.89	99.57	85.93	41.87 (57.9%)	31.08 (63.8%)
Salesman	1094.72	1075.05	85.69	72.39	92.78	79.38	42.64 (54.0%)	31.68 (60.1%)
Silent	1089.85	1068.37	94.65	80.58	108.60	95.05	42.22 (61.1%)	32.06 (66.1%)
Flower	3117.31	3058.69	290.99	253.52	416.04	377.29	159.19 (61.7%)	125.89 (66.6%)
Football	3586.41	3537.52	401.54	361.76	569.17	529.40	215.61 (62.1%)	182.09 (65.6%)
Hall	4391.42	1816.49	313.01	261.38	317.98	266.08	124.43 (60.9%)	87.46 (67.1%)
Highway	4622.77	4542.33	359.05	306.02	351.63	300.32	135.03 (61.6%)	98.30 (67.3%)
Mobile	4422.72	4337.27	464.82	408.56	632.26	575.63	213.00 (66.3%)	173.14 (69.9%)
News	4388.59	4317.52	340.10	288.51	343.09	291.54	127.33 (62.9%)	90.26 (69.0%)

Table 4.8: Speed gain of each algorithm with different reference frames

	reference = 2	reference = 5
Algorithm 1	not applicable	40∼50%
Algorithm 2	30∼40%	10∼20%
Algorithm 3	60∼70%	30∼40%

Table 4.9: Comparison of speed gain (in percent) with search range of ±32, number of reference frames is 5

	Kim [66]	Su [23]	Proposed
Carphone	21.52	51.4	66.9
Foreman	20.81	—	64.7
Mobile	20.45	37.7	69.9

algorithm reduces the motion estimation time approximately 60% to 75% with similar coding performance. Typically, the experiments show that the mean MET of the proposed algorithm was more than three times faster than the mean MET of the reference software. By reducing MET, the ET was correspondingly reduced significantly when the proposed algorithm was used. The contribution of each algorithm to the overall performance can be different for different test sequences.

When the most optimal reference frame is taken as the previous frame, the proposed algorithm does not require any subsequent selection. In our simulation, the major speed gain is from the initial reference frame selection algorithm (Algorithm 1) when five reference frames are used for ME. If ME is performed with less than two reference frames, the speed gain from Algorithm 1 is small.

The residue based mode decision (Algorithm 3) provides the largest contribution for the speed gain of ME with two reference frames. When the use of block mode is

appropriate, checking for sub-block mode is unnecessary. In our simulation around 60% to 70% of the speed gain is due to Algorithm 3 when ME is performed with two reference frames. The reminder of the speed gain is attributed to the early stop condition (Algorithm 2). Table 4.8 compares the speed gain of each algorithm when two and five reference frames are used for the proposed ME.

The residue based mode decision (Algorithm 3) provides the largest contribution for the speed gain of ME with two reference frames. When the use of block mode is appropriate, checking for sub-block mode is unnecessary. In our simulation around 60% to 70% of the speed gain is due to Algorithm 3 when ME is performed with two reference frames. The reminder of the speed gain is attributed to the early stop condition (Algorithm 2). Table 4.8 compares the speed gain of each algorithm when two and five reference frames are used for the proposed ME.

For comparison purposes, two examples of other work in the literature that report similar test parameters (number of reference frame, search range and quantization parameter, etc.) were selected. Table 4.9 compares the computational complexity of the proposed algorithm with that Kim, Han and Kim [66] and Su and Sun [23] in terms of MET. The results in Table 4.9 demonstrate an improvement in MET.

4.5 Summary

In this chapter, a fast ME algorithm for multiple reference frame coding in H.264/AVC is proposed. To reduce the complexity of ME, an initial reference selection algorithm, an estimation of the initial MV and early stop condition are considered together. Also, the difference between the current MB and reconstructed previous frames is used for fast mode decision. The performance of ME based on the JM13.1 reference

software and the proposed ME algorithm are compared in terms of PSNR vs. bit rate with the same compression conditions such as search range and search pattern. Furthermore, for the measurement of complexity, the proposed ME algorithm and the JM13.1 reference software are compared in terms of MET and ET.

From the simulation results, the proposed ME algorithm reduces MET significantly while the deterioration of PSNR can be considered to be minimal. The proposed ME algorithm has better performance for video sequences with small motion and static or slowing varying backgrounds. Since a fewer number of reference frames is required, and using variable block size, the proposed encoder reduces the complexity for ME. This situation is enhanced when there is high correlation between current and neighboring MBs.

Chapter 5

Distributed Video Coding

Distributed video coding has a different encoder structure from the conventional video encoder. In distributed video coding, the encoder performs only intra coding. Thus the complexity of the distributed video encoder is much less than the complexity of the conventional video encoder such as H.264/AVC. The temporal redundancy of the video sequence is reduced at the decoder side. The coding performance of the distributed video coding system is not as good as the performance of the conventional video coding system. Many methods have been introduced to improve the performance of the distributed video coding system. In this chapter, the theoretical background of distributed video coding systems and their performance are discussed. Also a proposed side information generation algorithm is introduced in this chapter. All figures and tables in this chapter and much of the text are based on previous work by the author [77].

5.1 Introduction

For efficient video coding, a substantial body of work has contributed to formulating international standards such as MPEG-1, 2 [78, 79], and the H.26x series [80] by the ITU-T [81]. Sikora [82] provides an overview of the MPEG-4 video coding standard. Rijkse [64] introduces the H.263 standards as an example of low bit rate communication. These video coding standards adopt both ME and transform coding to remove temporal and spatial redundancy. While spatial redundancies are reduced by transformation and quantization, motion estimation efficiently removes temporal redundancy existing between successive frames.

ME and transform coding require large computational complexity; it is a significant challenge in implementing video applications which require real time encoding. Chen *et al.* [75] and Huang *et al.* [65] tried to reduce the computational complexity of the H.264/AVC video coding system. Compared to encoder complexity, the complexity on the decoder is substantially less. Thus, in terms of complexity, the conventional video coding standards have an unbalanced encoder and decoder pair. The reason for the unbalanced encoder and decoder pair is that conventional video coding standards were designed for video applications having "one-time encoding and many-times decoding" such as in broadcasting and the entertainment industry. New video applications are emerging in many areas with restrictions for those applications demanding as many encodings as decodings. Also, encoders with low computational resources need new video coding schemes to reduce the complexity at the encoder. Video surveillance is an example requiring low complexity video encoding.

Distributed Video Coding (DVC) was introduced to reduce video encoder complexity. DVC is based on the Slepian-Wolf [3] and Wyner-Ziv [83] theories, which

were introduced in the 1970's. While the Slepian-Wolf theory provides a theoretical background for lossless compression, the Wyner-Ziv theory extends the Slepian-Wolf theory into the lossy compression area. Both the Slepian-Wolf and Wyner-Ziv theories specify the minimum boundary on the data rate when the correlation in source data is processed at the decoder instead of encoder. By removing the process for the removal of temporal redundancy, the encoder performs video coding with reduced computational complexity.

Girod *et al.* [2] adopted the Wyner-Ziv theory to implement low complexity video encoding and robust video transmission according to the structure depicted in Figure 5.1. This low complexity video encoder and decoder pair has become the *de facto* prototype for DVC. The major application area of DVC is real time video encoding and video encoding with low encoder resources. Puri *et al.* [37] proposed wireless sensor networks as an application of DVC. In wireless sensor networks the encoder resources may be limited. DVC has been used to implement robust data transmission in error prone wireless network channels which was proposed by Aaron *et al.*[39], and Puri *et al.* [84]. DVC is also useful in video surveillance applications, which consist of client and server side [40]. A backward channel aware DVC system was proposed by Liu *et al.* [85].

The DVC can be implemented in the compressed domain [86] as well as in the pixel domain [87]. Aaron *et al.* [88] showed that DVC in the compressed domain provides better performance than DVC in the pixel domain in most cases. The DCT is a widely used tool to transform from spatial domain data to frequency domain data because it reduces spatial correlation efficiently.

In DVC, the elimination of temporal redundancy, which requires the most com-

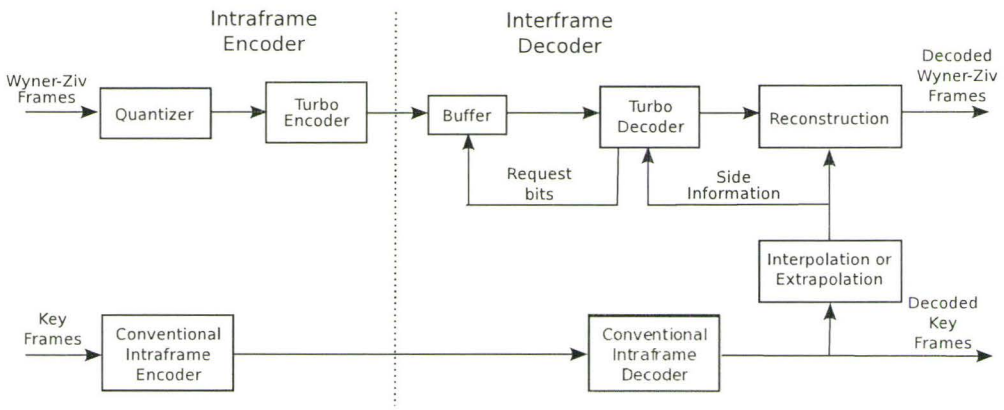


Figure 5.1: The structure of low complexity video encoder and decoder pair by Girod *et al.* [2].

putational complexity in video coding, is performed on the decoder side by adopting motion estimation. In DVC, the encoder performs only intra coding to remove spatial redundancy. In the encoder, as can be seen in Figure 5.1, each frame is classified as either a key frame or a Wyner-Ziv frame. For example, every even frame can be set as a Wyner-Ziv frame and every odd frame could be set as a key frame. Turbo coding which was proposed by Berrou and Glavieux [89] and *Low Density Parity Check* (LDPC) [90] codes are used for channel coding for Wyner-Ziv frames. Turbo coding applied to non-binary data compression has been extensively studied and a survey can be found in Frias' [91] work. Also Sartipi and Fekri [92] provided an example of LDPC coding for correlated sources at the decoder.

For the key frames, conventional intra video coding schemes which remove only spatial redundancy using DCT, quantization, intra prediction and entropy coding are used for encoding a frame. Instead of motion estimation at the encoder, both spatial and temporal redundancies are removed in the decoder in contrast with the

conventional video coding schemes. Although it is possible to regenerate Wyner-Ziv frames independently from key frames, using key frames in the decoder provides an advantage for regenerating the Wyner-Ziv frame. The information that is used in the procedure for regenerating the Wyner-Ziv frame is called *side information*. This procedure is called *side information generation* and plays a key role in improving the quality of reconstructed frames at the decoder.

In this paper, a new side information generation algorithm is proposed for a frequency domain DVC system. The proposed side information generation employs both multiple reference frames based motion estimation and post processing to improve the quality of the side information. A linear interpolation algorithm is adopted to find the motion vector for the current block. In the post processing procedure, minimum distortion block and residue based block selection algorithms are used to fix hole and overlapped areas in the side information frame.

This chapter is organized as follows. In Section 5.2 the background theory of DVC is briefly reviewed. Section 5.3 describes the Wyner-Ziv video coding system. In Section 5.4 a variety of side information algorithms are described. The proposed new side information generation algorithm is introduced in Section 5.5 and simulation results are given in Section 5.6. A summary is provided in Section 5.7.

5.2 Theoretical Background of Distributed Video Coding

DVC is a new video coding paradigm supporting intra encoding and inter decoding. Since it uses only intra encoding, distributed video coding reduces the encoder complexity significantly. Thus the target application is totally different from that of the conventional video coding standard. The origin of DVC theory is *Distributed Source Coding* (DSC). The word distributed refers to the encoders which are distributed and compress two statistically dependent signals X and Y independently without communication with each other. Examples of statistically dependent signals are correlated image and video sequences.

Let's assume there are two correlated information sources X and Y. In the conventional source coding, in terms of rate-distortion, the main issue is what is the minimum rate to code the sources X and Y when a joint encoder and joint decoder are adopted? The answer for this question is given by information theory:

$$\begin{aligned} R_X &\geq H(X) \\ R_Y &\geq H(Y) \\ R_X + R_Y &\geq H(X,Y) \end{aligned} \tag{5.1}$$

where $H(X)$ and $H(Y)$ are the entropy of sources X and Y, respectively, and $H(X,Y)$ is the joint entropy. Thus $H(X,Y)$ is the minimum achievable total rate when sources X and Y are coded and decoded without any error, i.e. lossless coding. Now consider the situation where there are two different encoders and one joint decoder for two correlated information sources X and Y. What is the minimum achievable total rate

for lossless coding? Slepian and Wolf provide the answer for this question.

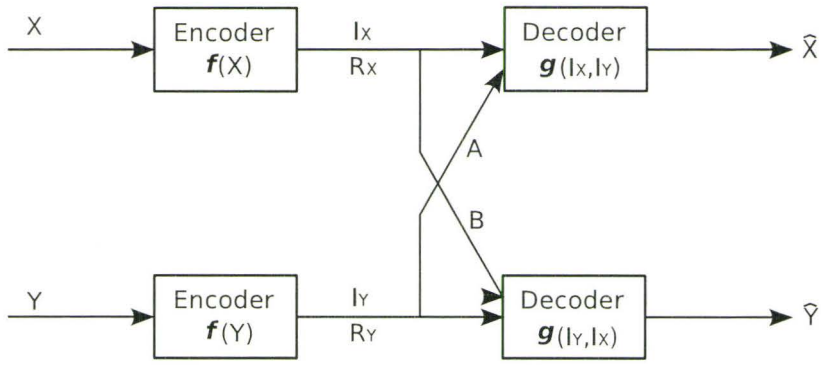
5.2.1 Slepian-Wolf Theory for Lossless Coding

The Slepian-Wolf theory specifies the minimum achievable rate to encode signals X and Y using an independent encoder and joint decoder. Consider two separate encoders and two joint decoders in a system depicted in Figure 5.2 (a). When the encoder f and the decoder g are used, then the encoded symbols for X and Y are $I_X \equiv f(X)$ and $I_Y \equiv f(Y)$, respectively. The size of symbols is represented with rate R_X and R_Y for I_X and I_Y, respectively. Then the decoded information is $\hat{X} \equiv g(I_X, I_Y)$ and $\hat{Y} \equiv g(I_Y, I_X)$. The error is defined by the difference between the original source X and reconstructed source $\hat{X}$, i.e. $\delta = d(X, \hat{X})$. For arbitrarily small error $\delta \geq 0$, the possible achievable rates are determined by the Slepian-Wolf theory, which is

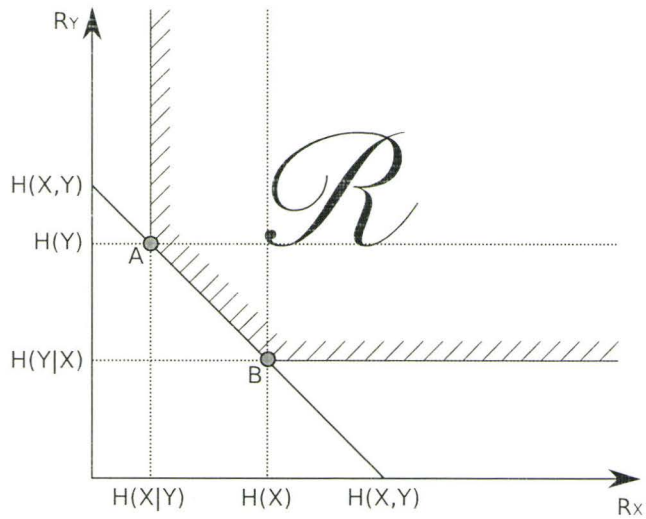
$$\begin{aligned} R_X &\geq H(X|Y) \\ R_Y &\geq H(Y|X) \\ R_X + R_Y &\geq H(X,Y) \end{aligned} \tag{5.2}$$

where $H(X|Y)$ is the condition entropy of X given Y and $H(Y|X)$ is the conditional entropy of Y given X. Equation 5.2 indicates that the rate of an independent encoder and joint decoder for dependent signals can be the same as the rate of the joint encoder and decoder pair.

Equations 5.1 and 5.2 show that when two correlated information sources are encoded independently with two encoders, the minimum achievable total bit rate is the same as for the case where the two sources are encoded with one joint encoder, and



(a)



(b)

Figure 5.2: The scenario of separate encoder and correlated decoder and its rate region [3].

the minimum achievable total bit rate is equal to the joint entropy. Figure 5.2 (b) depicts the achievable rate region for the separate encoder and joint decoder scenario. In Figure 5.2 (b), the hatched area, represented by the letter $\mathcal{R}$, is the achievable rate region when the system in Figure 5.2 (a) is implemented. In this case, the minimum achievable rate exists on the boundary lines.

5.2.2 Wyner-Ziv Theory for Lossy Coding

Let's consider again two correlated information sources X and Y with two independent encoders, one independent decoder and one joint decoder. Wyner and Ziv have studied this problem and provided the achievable rate in [83]. Thus the Wyner-Ziv theory is a special case of the Slepian-Wolf theory. For example, in Figure 5.2 (a), if channel B is opened and channel A is closed then source Y is encoded and decoded with an independent encoder and decoder pair, source X is encoded with an independent encoder and a joint decoder, and the minimum achievable rate point is the point A in Figure 5.2 (b) $(H(X|Y), H(Y))$. When channel A is open and channel B is closed in Figure 5.2 (a), then the minimum achievable rate point is the point B in Figure 5.2 (b) $(H(X), H(Y|X))$.

The Wyner-Ziv coding scenario is known as lossy compression with side information at the decoder. Wyner and Ziv considered that source Y is encoded and decoded without any loss while source X is compressed with a lossy scheme with acceptable distortion d. Then the Wyner-Ziv theory provides the minimum achievable

rate $R^{WZ}_{X|Y}(d)$, which is

$$R^{WZ}_{X|Y}(d) \geq R_{X|Y}(d), \qquad d \geq 0 \tag{5.3}$$

where $R_{X|Y}(d)$ is the minimum rate required to encode source X when source Y is available at both the encoder and decoder with average distortion d. If the distortion is zero ($d = 0$) the Wyner-Ziv theory corresponds to $R^{WZ}_{X|Y}(0) = R_{X|Y}(0)$, which is the Slepian-Wolf theory.

5.3 Wyner-Ziv Video Coding

Designing the DSC encoder and decoder pair to achieve the rate boundary of the Slepian-Wolf theory remains an open problem. Since both spatial and temporal redundancies are removed on the decoder side, the encoder is simple to design. Thus the performance of DSC mainly depends on how the decoder is designed. Although several efficient decoders have been made by Pradhan and Ramchandran [93], Aaron *et al.* [88] and Sartipi and Fekri [92], the rates are not close to the Slepian-Wolf's theoretical bound.

DVC is an example of the application of DSC. Due to the typically high correlation between successive video frames, video can be a good example for dependent signals X and Y. Figure 5.1 depicts a distributed video coding system based on the Wyner-Ziv theory, which was proposed by Girod *et al.* [2]. As we can see from Figure 5.1, the distributed video encoder consists of two independent intra coders: the Wyner-Ziv encoder and Key frame encoder. For the Key frame encoding, conventional intra frame codec is employed.

Wyner-Ziv frame coding can be done in either the pixel or transform domain. The DVC decoder consists of three parts: (1) key frame decoder, (2) Wyner-Ziv frame decoder and (3) side information generation. Usually the intra coding mode of conventional video coding standards is used in the key frame decoder. Wyner-Ziv frame coding can be performed with transform, quantization and different channel coding schemes. For example, Dalai *et al.* [94] implemented a turbo coding DVC system, and Satripi and Fekri [92] and Ascenso *et al.* [95] used LDPC coding to implement the channel coding in their DVC system. In the following subsections, two different Wyner-Ziv coders are discussed.

5.3.1 Pixel Domain Wyner-Ziv Frame Coding

First, each pixel value of the W-frame is quantized using a uniform scalar quantizer with 2^m levels. A quantized symbol stream q is sent to the Slepian-Wolf encoder, which consists of a turbo encoder and buffer (see Figure 5.1). At the decoder, for each W-frame, previously reconstructed K-frames are used to generate side information $\hat{S}$, which is an approximated version of W-frame W. The turbo decoder requests $k \leq m$ bits to reconstruct a quantized symbol $\hat{q}$. Then W-frame $\hat{W}$ is reconstructed and compared to the estimated W-frame $\hat{S}$. If the error between $\hat{W}$ and $\hat{S}$ is larger than the predefined threshold the turbo decoder requests another bit to buffer. Since the requested number of bits is less than the bits representing the pixel value compression is achieved. At the decoder, either an interpolation or extrapolation algorithm is adopted to generate side information $\hat{S}$.

5.3.2 Transform Domain Wyner-Ziv Frame Coding

Aaron *et. al* [86] implemented a transform-domain Wyner-Ziv video coder and compared the coding performance to that of a pixel domain Wyner-Ziv video coder. The structure of the transform-domain Wyner-Ziv video coding system is similar for the that of pixel domain except that an additional DCT block and a bit-plane extractor are added to the Wyner-Ziv encoder. First, the W-frame is transformed into a frequency domain. After quantization using a 2^m level quantizer m bit-planes are formed by the bit-planes extractor. Each bit-plane is encoded using a turbo encoder and stored at the buffer. At the decoder side, the interpolated or extrapolated side information is transformed using the same DCT coder, and extracted bit-plane quantized coefficients.

5.4 Side Information

For Wyner-Ziv frame decoding, high quality approximation of unknown data for the current Wyner-Ziv frame is desirable for an efficient rate and good performance. Side information provides an approximated Wyner-Ziv frame to help reconstruction of the Wyner-Ziv frame at the decoder. Side information plays a key role in DVC [41] decoding and thus generating high quality side information is essential to achieve high performance. Several side information generation algorithms have been proposed by Lu *et al.* [41], Li and Delp [43] and Wang *et al.* [96]. To generate side information in the decoder, the most popular methods are interpolation and extrapolation. Both interpolation and extrapolation are performed using reconstructed key frames, which are encoded and decoded with an intra coding scheme, in the encoder and decoder,

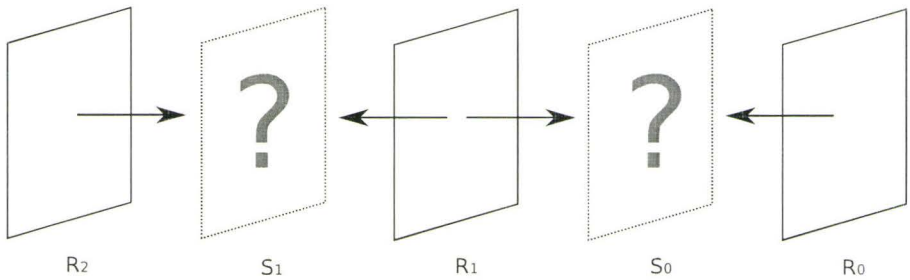


Figure 5.3: Generating side information using interpolation method. The side information (frame with dotted line) can be generated using both previous and future frames (frame with solid line).

respectively. Interpolation and extrapolation algorithms are briefly discussed in the following subsections.

5.4.1 Interpolation for Side Information

Interpolation is a widely used side information generation method in the distributed video coding system because of its efficiency and simplicity. To generate a frame using the interpolation method, both previous and future frames are required. Interpolating a frame is an approximation based on the nearest neighboring frames, which is

$$\hat{S}_i = \mathcal{I}(S_i|R_i, R_{i+1}), \tag{5.4}$$

where $\mathcal{I}()$ is an interpolation function, and R_{i+1} and R_i are previous and future frames, respectively, which are already reconstructed. Figure 5.3 shows the structure of the interpolation method in side information generation. In Figure 5.3, there is unknown knowledge for each piece of side information S_i at the decoder side. Thus each piece of side information should be interpolated using neighboring frames. If

the linear interpolation algorithm is adopted to interpolate the side information, the pixel value at each position (x, y) is

$$S_i(x, y) = \left\lfloor \frac{R_i(x, y) + R_{i+1}(x, y)}{2} \right\rfloor \tag{5.5}$$

where $R_i(x, y)$ is the pixel value at position (x, y) in reference frame R_i. Other interpolation algorithms can be adopted to generate better side information.

5.4.2 Extrapolation for Side Information

Extrapolation is an another method to generate side information using previously generated K-frames. While interpolation requires a future frame, previous frames are enough to generate side information in extrapolation. The overall structure of extrapolation is shown in Figure 5.4. In Figure 5.4, the side information frame S_i is unknown at the decoder side. The extrapolation process with K-frames can be expressed mathematically as

$$\hat{S}_i = \mathcal{E}(S_i | R_i, R_{i+1}, \cdots) \tag{5.6}$$

where $\mathcal{E}()$ is an extrapolation function. For example simple linear extrapolation can be used to reconstruct S_i using two reconstructed frames R_i and R_{i+1} as

$$S_i(x, y) = R_i(x, y) + (R_i(x, y) - R_{i+1}(x, y)) . \tag{5.7}$$

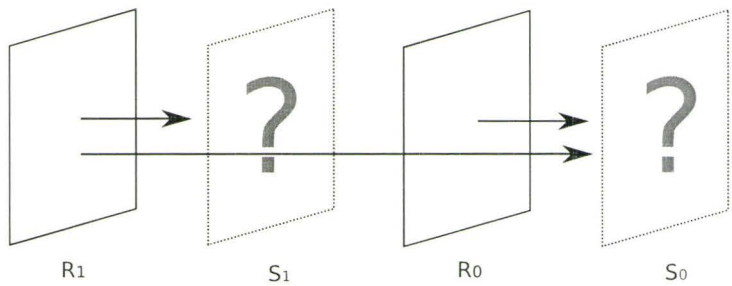


Figure 5.4: Generating side information using extrapolation method. The side information (frame with dotted line) can be generated using previous reconstructed frames (frame with solid line).

Also the extrapolation process can be done with both previous frames and side information frames, which is

$$\hat{S}_i = \mathcal{E}(S_i|R_i, S_{i+1}, R_{i+1}, S_{i+2}, \cdots). \tag{5.8}$$

To estimate each side information frame, the extrapolation function needs to investigate the relationship among previously reconstructed frames.

5.4.3 Motion Estimation for Side Information

The accuracy of side information can be improved by combining interpolation with other techniques, such as motion estimation. Although motion estimation requires high computational resources, it is widely used in video coding because it removes temporal redundancy, efficiently. In the conventional video coding standards, motion estimation is used only in the encoder [66]. Motion estimation can be used to generate side information for DVC because there is high temporal redundancy between successive frames. To improve performance, motion estimation and interpolation (or

Table 5.1: Comparison of the encoding time of different encoders (all times given in milliseconds)

Sequence		Inter	Intra	DVC	Speed gain(%)	
					SG_{Inter}	SG_{Intra}
QCIF (176×144)	**Carphone**	514.75	158.00	52.00	89.9	67.1
	Container	519.75	155.00	51.75	90.0	66.6
	Foreman	520.00	157.00	64.25	87.6	59.1
	Salesman	517.75	190.25	58.50	88.7	69.3
CIF (352×288)	**Hall**	2000.00	547.25	290.25	85.5	47.0
	Mobile	2111.00	658.00	298.00	85.9	44.7
	News	2006.75	588.00	329.00	83.6	44.0
	Stefan	2100.75	600.25	286.50	86.4	42.3

extrapolation) are often combined. Interpolation is adopted to estimate motion vectors as well as the pixel values of approximated frames.

5.5 Improved Side Information Generation

The complexity of a video encoder can be measured by its encoding time. Table 5.1 shows the encoding time of the H.264/AVC inter coding, the H.264/AVC intra coding and DVC. For the H.264/AVC inter coding, one reference frame was used for motion estimation. In Table 5.1 speed gains SG_{Inter} and SG_{Intra} are calculated as

$$\begin{aligned} SG_{Inter} &= \frac{ET_{Inter} - ET_{DVC}}{ET_{Inter}} \times 100(\%), \\ SG_{Intra} &= \frac{ET_{Intra} - ET_{DVC}}{ET_{Intra}} \times 100(\%) \end{aligned} \tag{5.9}$$

where ET_{Inter}, ET_{Intra} and ET_{DVC} are encoding times of inter, intra and DVC coding, respectively.

Using the DVC for the CIF video format encoding, the encoder complexities

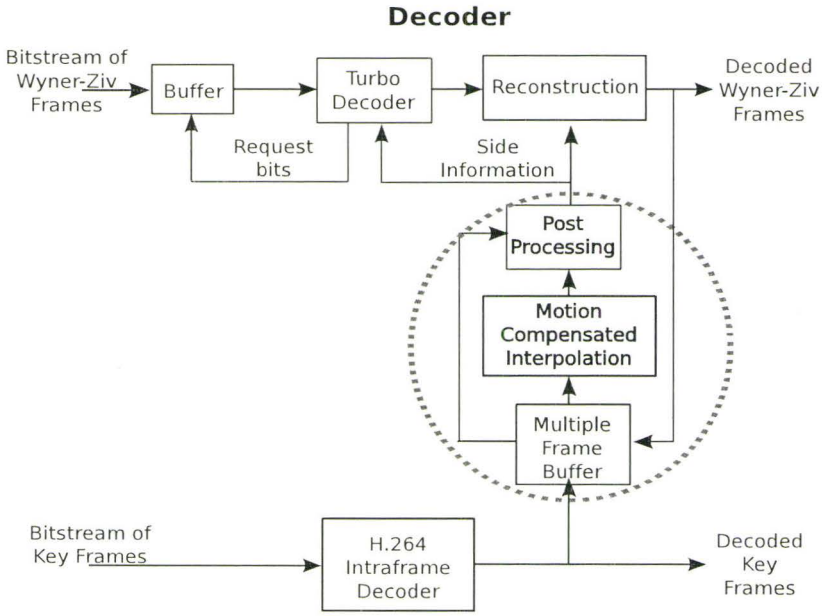


Figure 5.5: The decoder using proposed side information generation with multiple reference frames motion estimation and post processing.

are decreased by 85% and 45% compared to conventional H.264/AVC inter and H.264/AVC intra encoding, respectively. When the video frame format is QCIF the advantage is more evident. The reduced complexities are 89% and 65% compared to conventional H.264/AVC inter and H.264/AVC intra encoding, respectively. From these simulation results, we can see that DVC contributes to reduce encoding complexity of the video encoder significantly.

It is widely known that the performance of DVC is typically below that of conventional video coding with motion estimation (inter coding) in the encoder. The main reason for the low performance of DVC is that it does not remove temporal redundancy on the encoder side. Also, channel coding such as turbo coding and LDPC increases the bit rates. Since Wyner-Ziv frames are decoded using side information to

improve the coding efficiency, high quality side information is mandatory. Although previous side information generation algorithms provide improved performance for DVC, the achievable rate is still far from the Wyner-Ziv boundary. We propose a novel side information generation algorithm based in the decoder as shown in Figure 5.5 (circled block). In the decoder, the side information generation block consists of three parts. These are: (1) multiple reference frames based motion estimation, (2) motion compensated interpolation, and (3) post processing. Algorithm 5 describes the proposed DVC decoding process using improved side information.

For multiple reference frames based motion estimation, a frame buffer is added into the side information generation block. Multiple reference frames based motion estimation is similar to that used by the H.264/AVC video encoder. Previously decoded key frames and Wyner-Ziv frames are used as reference frames for the motion estimation of the current frame. Various motion vector search algorithms such as hexagon search proposed by Zhu *et al.* [14], diamond search proposed by Zhu and Ma [12] and three step search proposed by Koga *et al.* [6] are available for fast motion estimation. In the proposed side information generation, full search motion estimation is used for a high quality approximation. More details about motion compensated interpolation and post processing are described in the following subsections.

5.5.1 Interpolation Based on Multiple Reference Frames Motion Estimation

Although pixel-based interpolation or extrapolation for side information generation provides acceptable performance, Li and Delp [43] show that the motion estimation based algorithm achieves better performance. Since the computational complexity of

Algorithm 5 Proposed side information generation algorithm

Read the current Wyner-Ziv stream
Generate side information frame
for all macroblocks **do**

(1) Find reference frame and motion vectors satisfying minimum distortion.

(2) Construct a side information frame using reference number and motion vectors.

(3) Post processing for the hole and overlapped area.

end for
Reconstruct Wyner-Ziv frame
for all 8×8 blocks **do**

(1) Read side information corresponding to the current block.

(2) Perform DCT, quantization, and zig-zag scan.

(3) Turbo encode the side information block.

(4) Form turbo decoder input stream using parity bits and the output of the turbo encoder of side information.

(5) Turbo decoding.

(6) Improve the reconstructed frame using side information frame.

Choose the best macroblock mode for the current macroblock.
end for

motion estimation linearly increases in proportion to the number of reference frames and the number of search points, a single reference frame is widely used for motion estimation. When large motion and occlusion exist in a frame, motion estimation with one reference frame does not provide enough information to approximate the unknown side information frame. Thus it is proposed to take multiple reference frames for motion estimation to generate high quality side information frames. For each block, only one frame is chosen from multiple reference frames, as follows.

Using motion estimation, after obtaining the motion vectors and residuals from multiple reference frames, generating side information for each block can be generated from:

$$\hat{S}_i = \mathcal{A}\begin{pmatrix} S_i|R_{i-1}, R_{i+1}, R_{i+2}, \cdots, R_{i+n}; \\ MV_{i+1}, MV_{i+2}, \cdots, MV_{i+n}; \\ d_{i+1}, d_{i+2}, \cdots, d_{i+n} \end{pmatrix} \tag{5.10}$$

where $\mathcal{A}()$ is an approximation function which is implemented with two steps:

(1) finding a frame number index having the smallest residual such as

$$k = \arg\min_i d_i \qquad 1 \leq i \leq n. \tag{5.11}$$

(2) interpolating each block in the side information as

$$\hat{b}_{1,\Delta x,\Delta y} = \mathcal{I}\left(b_{1,\Delta x,\Delta y}|b_{0,x,y}, b_{k,x+mv_x,y+mv_y}\right) \tag{5.12}$$

where b_1, b_0 and b_k are the part of the current frame R_1, previous frame R_0 and the kth frame R_k, which has minimum residue, respectively, and mv_x and mv_y are motion vectors in the horizontal and vertical directions, respectively.

By using linear interpolation, the displacement (Δx and Δy) can be obtained from

$$\begin{aligned} \Delta x &= x + \frac{mv_x}{k} \\ \Delta y &= y + \frac{mv_y}{k}. \end{aligned} \tag{5.13}$$

Then, the approximated block is

$$b_{1,\Delta x,\Delta y} = \frac{(k-1) \times b_{0,x,y} + b_{k,x+mv_x,y+mv_y}}{k}. \tag{5.14}$$

The approximated block is placed at the position Δx and Δy of the approximated side information frame. Since each block of the approximated side information frame is copied and pasted from the reference frame at the new position $(x + \Delta x,\ y + \Delta y)$, the approximated side information frame is not perfect. Overlap as well as holes can exist on the approximated side information frame, as shown in Figure 5.6.

5.5.2 Post Processing for Improved Side Information

After motion estimation and compensation, there is a high probability that holes and/or overlapped areas on the generated side information frame will exist. Figure 5.6 shows an example. In Figure 5.6, p_1 is an example of a hole area surrounded by blocks B_1, B_2, B_3 and B_4, and p_2 is an example of an overlapped area by blocks B_5 and B_8. Post processing of the reconstructed side information frame contributes to improving the quality of the generated side information frame. For the hole area,

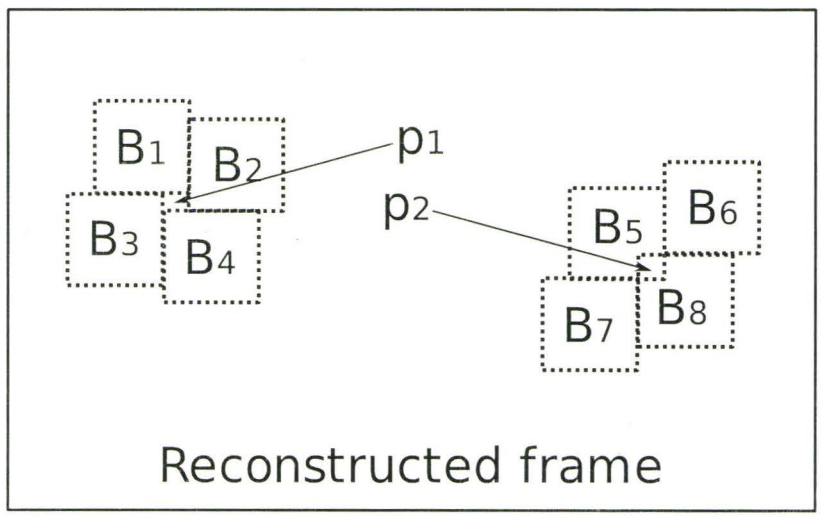


Figure 5.6: The example of hole and overlapped area after side information generation.

if the size of the hole area is more than four pixels, motion estimation is repeated to copy and paste from a reference frame. Otherwise a filtering algorithm such as median or mean filtering using the neighboring pixel values can be employed. For its simplicity, mean filtering is adopted in our proposed algorithm.

The overlapped area is recovered by comparing the residues of overlapping blocks such as

$$\mathsf{O}_k = \begin{cases} \mathsf{B}_i & \text{if } d_i \leq d_j \\ \mathsf{B}_j & \text{otherwise} \end{cases} \tag{5.15}$$

where O_k is the kth overlapped area, and d_i, d_j are residuals of block B_i and B_j, respectively. Figure 5.7 shows an example of a generated side information frame in the decoder. Figure 5.7 (a) and (b) are the original 8th frame of the **Carphone** sequence and corresponding generated side information frame, respectively. Although there is

some blurring around the head area due to motion, the quality of the generated side information is enough to be used for re-constructing the Wyner-Ziv frame.

5.5.3 Decoding Wyner-Ziv Frame Using Side Information

After generating the side information frame, the decoder performs DCT, quantization and turbo encoding for all blocks in the side information frame. Then, the parts of the output of the turbo encoder are combined with the bits from the encoder buffer. The bit array feeds into the turbo decoder to decode the Wyner-Ziv frame. Dequantizing and inverse DCT are performed on the output of the turbo decoder. After decoding, the distortion between the decoded Wyner-Ziv frame and the side information frame is measured. If the distortion exceeds a threshold, the decoder requests more bits from the encoder buffer and performs the decoding procedure.

5.6 Simulation Results and Discussion

In this section, the DVC performance based on the proposed side information generation algorithm is evaluated. For key frame coding, the intra coding mode of the H.264/AVC reference software JM 13.1 [76] was used. For testing, four CIF format sequences and four QCIF sequences were used. The test sequences and their characteristics are described in Table A.1. For different bit rates, variable quantization parameters are used for both H.264/AVC intra and inter frame coding. For intra frame coding, the quantization parameters vary from 22 to 32. For inter frame coding, the quantization parameter ranges from 20 to 28. In inter frame coding one

(a)

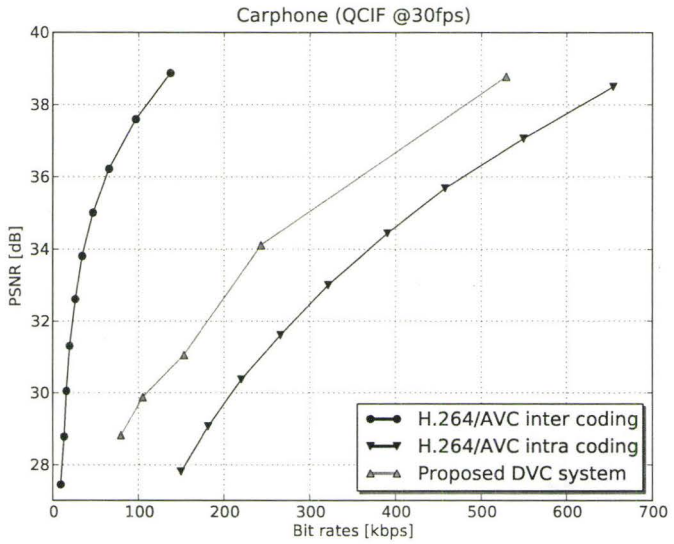
(b)

Figure 5.7: Generated side information frame after motion estimation using multiple reference frames and post processing: (a) original and (b) reconstructed side information frame.

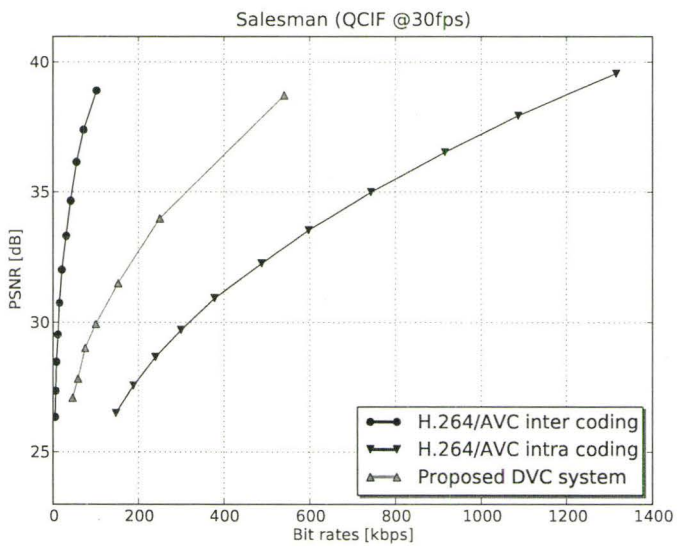
reference frame is used for the motion estimation at the encoder. The search range for motion estimation is fixed to ± 16. CAVLC is used to compress the bitstreams for both intra and inter coding.

For DVC, the key frame coding block is implemented using conventional H.264/AVC intra frame coding. The DCT is adopted to transform a Wyner-Ziv frame at the encoder. H.264/AVC Intra quantization matrix is used for the DCT coefficients, then turbo coding is implemented to generate the bitstream from the quantized coefficients. After turbo coding, the bitstreams are saved in the frame buffer. In the decoder, the side information frame is constructed using motion estimation with a maximum of five reference frames using full search. The block size and search range for motion estimation are 8 and ± 16, respectively. Since the DVC does not perform intra prediction or inter prediction, its encoder complexity is much less than the encoder complexity of the H.264/AVC intra encoder as seen in Table 5.1. As mentioned earlier, low encoder complexity is a main advantage of a distributed video encoder.

Figures 5.8 and 5.9 show the PSNR vs. bit rates of three different coding algorithms when QCIF format test sequences are used. The test sequences are: **Carphone**, **Salesman**, **Foreman** and **Container**. The quantization parameters are chosen for the target bit rates between zero to 1.4 Mbps. In the **Carphone** sequence (Figure 5.8 (a)), the performance of the proposed DVC is lower than the performance of H.264/AVC inter coding. At high bit rates, the difference is significant but still better than the performance of the H.264/AVC intra coding. Figures 5.8 (b), and Figures 5.9 (a) and (b) shows a similar pattern of rate distortion curves when **Salesman**, **Foreman** and **Container** are used as the test sequence. In those sequences, the performance of the H.264/AVC inter coding is outstanding compared to the performance

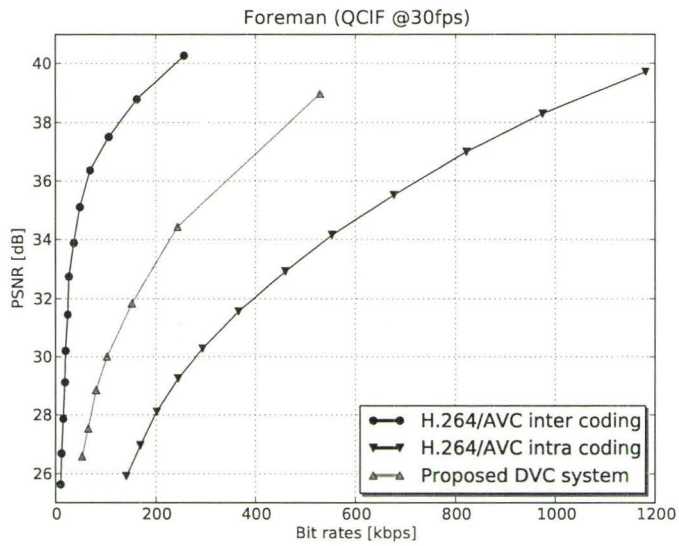


(a)

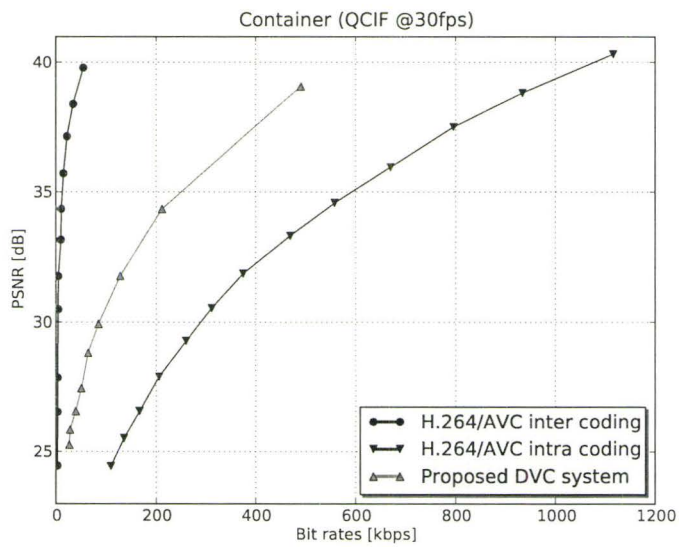


(b)

Figure 5.8: Comparison of the coding efficiency of the proposed algorithm with the conventional video coding algorithm. The test sequences are QCIF format: (a) **Carphone** and (b) **Salesman** sequences.



(a)



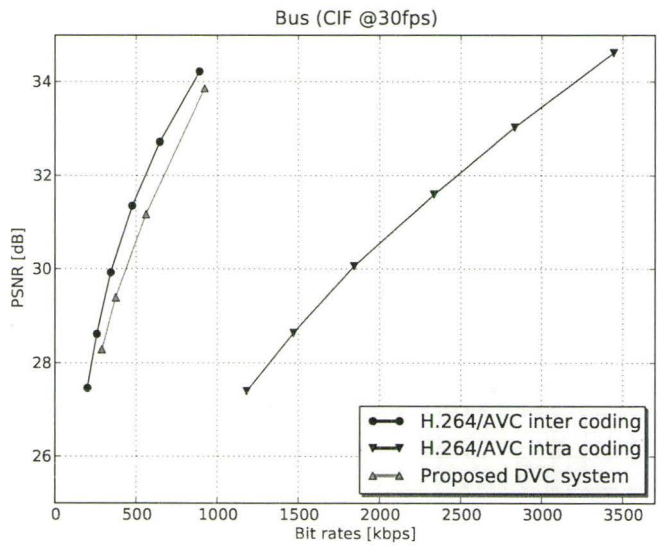
(b)

Figure 5.9: Comparison of the coding efficiency of the proposed algorithm with the conventional video coding algorithm. The test sequences are QCIF format: (a) **Foreman** and (b) **Container** sequences.

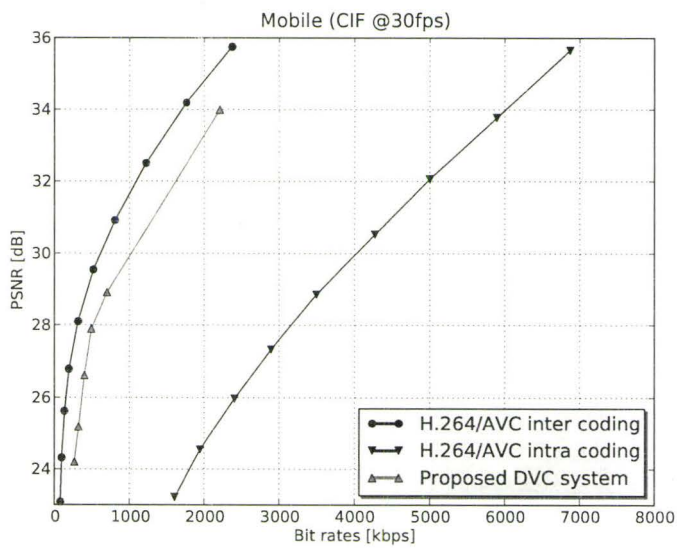
of the H.264/AVC intra coding and the proposed DVC. Although the performance of the proposed DVC is worse than the performance of the H.264/AVC inter coding, it demonstrates improved performance over the H.264/AVC intra coding with low encoder complexity. In all the test sequences, the performance of the proposed DVC is located between the performance of the intra and inter codings.

Figures 5.10 and 5.11 show another example of coding performance of the proposed DVC when CIF test sequences are used. The test conditions are similar to that of the QCIF test sequences. Figure 5.10 (a) shows rate-distortion curves of three different coding modes for the **Bus** sequence. Although the performance of the proposed DVC is worse than the performance of inter coding, the performance of the proposed DVC is much better than intra coding for all bit rates. For the case of the **Mobile** sequence, the coding performance of the proposed DVC is similar to the performance of the inter coding and much better than the performance of intra coding.

Figure 5.11 (a) depicts the rate-distortion curve for the **News** sequence. Although the coding performance of the proposed DVC system is lower than that of inter coding, it is better than intra coding. Throughout various test sequences, the proposed DVC system provides good coding performance situated between the H.264/AVC inter and intra coding. Figure 5.11 (b) represents the rate distortion curve of the **Football** sequence. The pattern of the proposed DVC performance is similar with that of the H.264/AVC inter coding. As shown in Figure 5.10 (b) and Figure 5.11 (b) the performance of the proposed DVC is comparable to the performance of the H.264/AVC inter coding and much better than the performance of the H.264/AVC intra coding.

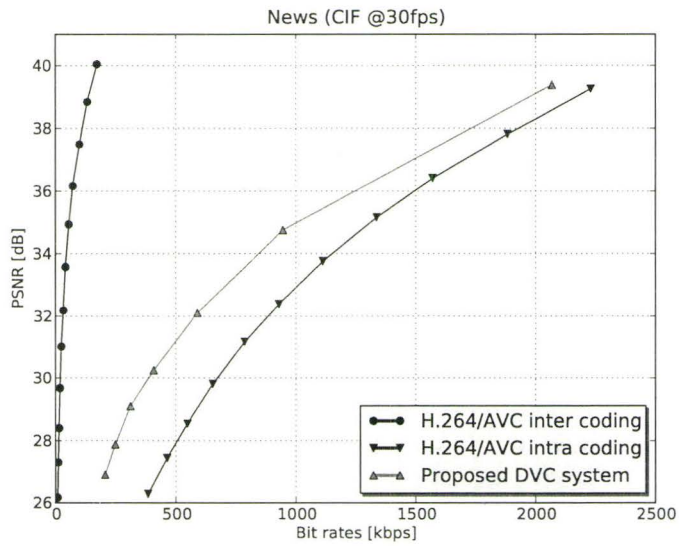


(a)

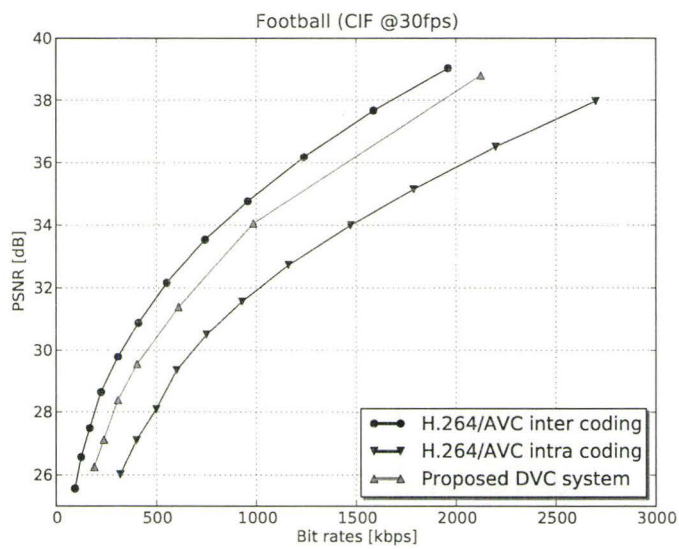


(b)

Figure 5.10: Comparison of the coding efficiency of the proposed algorithm with the conventional video coding algorithm. The test sequences are CIF format: (a) **Bus** and (b) **Mobile** sequences.



(a)



(b)

Figure 5.11: Comparison of the coding efficiency of the proposed algorithm with the conventional video coding algorithm. The test sequences are CIF format: (a) **News** and (b) **Football** sequences.

The encoding complexity of the proposed DVC system is significantly lower compared to conventional video coding standards such as H.264/AVC. For the CIF video format encoding, the encoder complexities can be reduced by 85% and 45% compared to conventional H.264/AVC inter and H.264/AVC intra encoding, respectively. For the QCIF format, the reduced complexities are 89% and 65% compared to conventional H.264/AVC inter and H.264/AVC intra encoding, respectively.

5.7 Summary

In this chapter, a new video coding scheme, which is called distributed video coding is introduced for low complexity video encoder. The theoretical background as well as different encoding and decoding methods for distributed structure are discussed. Since distributed video coding adopts independent frame coding at the encoder, the coding efficiency of distributed video coding is lower than the conventional video coding standard with inter coding. To improve the coding efficiency, distributed video coding eliminates temporal redundancy at the decoder side. Side information frames provide the number of required bits as well as the approximated pixel values. Improving the quality of the side information frame is critical in distributed video coding to achieve high coding efficiency.

By designing a new side information generation algorithm, the coding efficiency of distributed video coding is improved. Both multiple reference frames based motion estimation and post processing are adopted in the proposed side information generation algorithm. Motion estimation using multiple reference frames approximates the current Wyner-Ziv frame when there is large motion on the object and complicated backgrounds on a frame. Simple mean filtering works well for recovering hole areas on

the side information frame. The residue based overlapped block selection algorithm improves the quality of the regenerated side information frame. The encoding complexity of distributed video coding is reduced by more than 30%. The performance of DVC using the proposed side information algorithm is improved compared to the conventional standards.

Chapter 6

Conclusions, Remarks and Future Work

In this thesis, a wide range of video coding standards and their performance have been presented. In particular, the complexity issue of the video encoder is discussed in detail. An efficient ME algorithm for a low complexity video encoder was proposed. Also side information generation algorithms for distributed video coding system were explained. This chapter provides a summary of this thesis as well as conclusions and future work. This chapter is organized as follows: In Section 6.1 we summarize this thesis briefly. Concluding remarks about the current research are provided in Section 6.2. This thesis is finalized with future work, which is related to the current research in Section 6.3.

6.1 Summary Review

This section provides a summary of this thesis. The concept of a low complexity video encoder is introduced with different applications in Chapter 1. Previous work on efficient ME and distributed video coding algorithms are discussed in Chapter 1. Also Chapter 1 describes video coding tools such as ME, transformation, quantization, and entropy coding as well as the performance measure tool. Chapter 2 introduces various international video coding standards such as MPEG-1, MPEG-2, MPEG-4, H.261, H.263 and H.264. Key functions of each standard are explained in detail. Although there are improvements in coding performance the later video coding standard has greater computational complexity than the earlier video coding standard because it adopted more techniques to remove temporal and spatial redundancies. The coding performances of video coding standards are compared and discussed in Chapter 2.

The computational complexity of video encoder algorithms is studied in Chapter 3. Since ME consumes the most of the encoder complexity, the complexity of ME is discussed in detail. Various efficient ME algorithms for low complexity video encoders have been proposed. Different search pattern algorithms are examples of methods that have contributed to reducing the complexity of video encoders. H.264 video encoder adopts both multiple reference frames based ME and variable block size mode decisions to improve coding efficiency. Both techniques increase the computational complexity of the ME block.

In Chapter 4, an efficient ME algorithm is proposed to reduce encoding time and ME time. The information from neighboring blocks is used to determine the initial reference frame, initial motion vector and the thresholds for an early stop condition. The coding performance and the complexity of the proposed efficient ME algorithm

is discussed in Chapter 4.

Chapter 5 introduces a distributed video coding structure, its differences from conventional video coding structures, and the theoretical background and several applications. Side information plays a key role in the distributed video coding system because it provides temporal redundancy information for reconstructing video frames. A new side information generation algorithm is proposed to improve the coding performance of the distributed video coding system. In the proposed algorithm multiple reference frames based ME and post processing are adopted. The coding performance of the distributed video coding system using the proposed side information generation algorithms is compared to the performance of H.264 at the end of Chapter 5.

6.2 Conclusions and Remarks

Several conclusions can be drawn from this work. The conventional video coding standards have too heavy computational complexity to implement real time applications. To reduce the encoder complexity, optimization of function blocks in the video encoder is required. A low complexity video encoder can be achieved by designing efficient ME. When there is high spatial correlation among neighboring blocks, the information from the neighboring blocks is useful to estimate an initial reference frame, initial motion vector and early stop decision. Residual information can be used for a fast mode decision algorithm in H.264 video encoder.

Distributed video coding provides a low complexity encoding system compared to conventional video coding standards. Low complexity of the distributed video encoder is made possible by performing intra only coding. Although the Slepian-Wolf and Wyner-Ziv theories provide the lower boundary of achievable rate-distortion curves,

the performance of the current distributed video coding system is far from the lower boundary. The coding performance of the propose distributed video coding system is substantially lower than that of H.264 inter coding. To improve the coding performance designing a smart decoder is required. The side information block plays a key role in distributed video decoding. This research contributes to generating high quality side information for distributed video coding. In the following subsection, the contributions of the proposed efficient ME algorithms for H.264 video encoder and improved side information generation algorithm for distributed video decoding are described.

6.2.1 Discussion of Results of Efficient Motion Estimation

A low complexity video encoder can be achieved by designing an efficient ME algorithm. Since H.264 supports both multiple reference frames based motion estimation and variable modes for each MB, the H.264 video encoder has greater computational complexity than other video coding standards. Neighboring MBs provide enough information for initial reference frame selection, initial MV and thresholds for early stop condition. While early stop conditions contribute somewhat to a speed gain, the initial reference selection algorithm gives a major speed gain for ME.

The proposed residue based mode decision algorithm provides reduced motion estimation time while maintaining coding performance. Since checking sub-block mode is not required most of the time, residue based mode decision provides the largest contribution for the speed gain of ME. The proposed ME algorithm reduces the MET approximately 60% to 75% with similar coding performance compared to the conventional H.264 video encoder with full search and EPZS search pattern.

6.2.2 Discussion of Results of Improved Side Information for DVC

The DVC system reduces encoder complexity significantly compared to conventional video coding standards. The encoding time of DVC is just 10% to 20% of the encoding time of the H.264 video encoder. Although the DVC system has low encoder complexity, it has not been widely adopted for real time applications. The low coding performance of the DVC system is the main obstruction for implementing real time applications. A DVC system can be designed in both the pixel domain and transform domain. Usually the transform domain based DVC system has better performance than the pixel domain based DVC system. DCT is widely employed in transformation. Turbo coding and LDPC are the coding scheme chosen here to implement channel coding in the DVC system. Much research has been done to improve the coding performance. Since side information plays a key role to reconstruct the Wyner-Ziv frame at the decoder, that research was focused on improving the quality of side information.

The proposed DVC system is implemented using DCT for transformation and turbo coding for channel coding. In the proposed side information generation algorithm, multiple reference frames are used for ME followed by post processing. The linear interpolation algorithm works well to estimate MV as well as pixel values after ME. Simple mean filtering is effective for recovering hole areas on the side information frame. The residue based block selection algorithm improves the overlapped area on the side information frame. The performances of the DVC system with the proposed side information generation algorithm are between that of H.264 intra coding

and H.264 inter coding. Depending on test sequences, the performance of the proposed DVC system approaches the performance of H.264 inter coding while requiring reduced computational complexity.

6.3 Future Work

The area of video coding has many applications required from low complexity encoder/low complexity decoder to high complexity encoder/high complexity decoder for high coding performance. The video coding algorithm should be optimized for target applications. Different video coding strategies can be proposed and implemented for different applications. This work provided some algorithms for low complexity video encoders. However, there are numerous opportunities for future work in the video coding area.

6.3.1 Designing Low Complexity Video Encoder and Decoder Pairs

The main advantage of the DVC system is its low encoder complexity. The complexity of the decoder side of the DVC system is significantly higher compared to the decoder of conventional video coding standards. Most complexity is from turbo coding because it has an iterative recursive structure. Thus, the proposed low complexity encoder and low complexity decoder system has a DVC encoder and conventional video decoder structure. In those systems, the central server should perform trans coding. The trans coder takes the DVC bitstream and generates a conventional video coding bitstream. In that case, the central server has huge computational complexity.

Also there are latencies in communication between encoder and decoder.

Designing new video coding standards is still in progress by international groups such as MPEG and ITU-T. But the objective of those standards is not much different from that of the current video coding standards. Designing a video coding system with a low complexity video encoder and decoder is challenging research work. Many ways exist for designing a video coding system, which is totally different from the conventional video coding standards. For a low complexity encoder, different coding tools can be tested and adopted. For example, there are different transformation methods for video signal. Finding an efficient method to remove temporal redundancy is another way to achieve a low complexity video encoder.

6.3.2 Future Implementation Work

Video coding standards provide reference encoder and decoder software. Implementing the video encoder in the software level optimizes the reference encoder to adapt to a specific application. The efficient ME algorithm is a good example of implementation of optimized video encoder. The current work is related with building a low complexity video encoder using software implementation. The advantage of software implementation is that it is easy checking mistakes in the code, and verifying the performance. However, hardware implementation has many advantages over software implementation.

For hardware implementation, power consumption, speed and complexity should be considered together. Some trials have been done to implement video codec on *field programmable gate array* (FPGA) chips and *very large scale integration* (VLSI)

architecture. Usually, a video codec is implemented using both software and hardware together to use the merit of each part. Generally hardware implementation is better than software implementation in terms of power consumption and execution speed. But software implementation has more flexibility than hardware implementation. Thus computationally complex parts such as ME and transformation are implemented using hardware and the other parts can be implemented using software. Since there are trade offs between software and hardware, the selection is the designer's decision. Some implementation issues of the H.264 encoder for broadcast applications can be found in [97].

Hardware implementation requires other knowledge besides video coding standards such as system architecture, memory allocation, multi threads coding, etc. Parallel processing is one advantage of a hardware implementation. By using parallel processing, the encoding time and decoding time can be reduced significantly. *Very high speed integrated circuits* (VHSIC) were introduced in 1980's. Later *VHSIC hardware description language* (VHDL) was widely used for design FPGA and application specific integrated circuits (ASIC). Comparing with software implementation, hardware implementation provides the advantage in speed and disadvantage in maintenance.

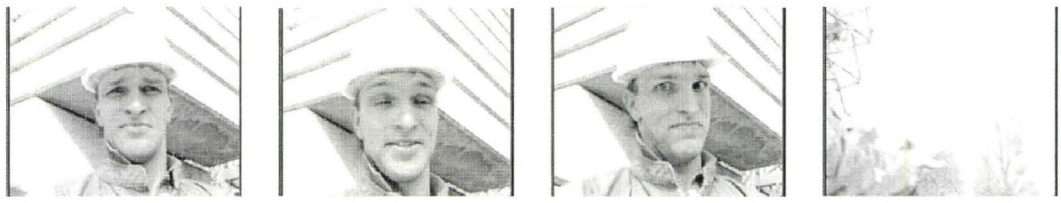
Appendix A

The List of Test Sequences

The test sequences which are provided by international standard groups are summarized in Table A.1. Those test sequences are used for the simulations in this thesis. Some selected frames of the test sequences are shown in Figure A.1 - Figure A.4. The test sequences in Figures A.1 and A.2 are QCIF format. The test sequences in Figures A.3 and A.4 are CIF format.

Table A.1: Video test sequences and characteristics of each sequence

Format	Sequence	Frames	Characteristics
QCIF (176×144)	**Carphone**	300	Large motion of a big object, part of background has large motion.
	Container	300	Small motion of a big object, simple background.
	Foreman	300	A big object with motion and panning.
	M&D	300	Two objects with small motion, the background is static.
	Salesman	300	Small motion of a big object, detailed background with small motion.
	Silent	300	A big object with motion, the background is static.
CIF (352×288)	**Bus**	150	A large object moving with fast motion.
	Flower	250	Small motion, detailed background.
	Football	260	Many objects with fast and large motion.
	Hall	300	Two or three persons walking on office hall.
	Highway	300	Fast motion of object and background.
	Mobile	300	Complicated motion of objects and moving calendar.
	News	300	Two objects in foreground with small motion and two objects in the background with large motion.

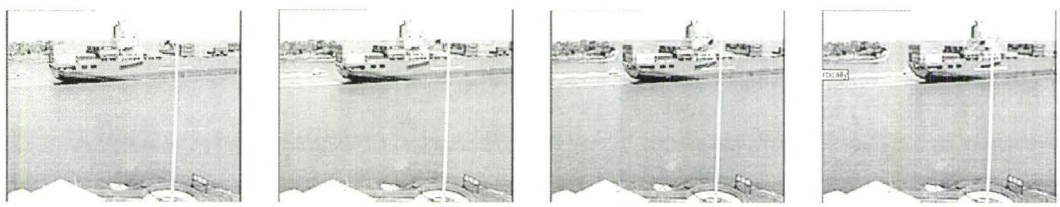


#000 #100 #200 #300

(a)

#000 #100 #200 #300

(b)



#000 #100 #200 #300

(c)

#000 #100 #200 #300

(d)

Figure A.1: Sample frame numbers from QCIF format test sequences: (a) **Foreman**, (b) **Carphone**, (c) **Container** and (d) **Salesman**.

#000 #100 #200 #300

(a)

#000 #200 #400 #600

(b)

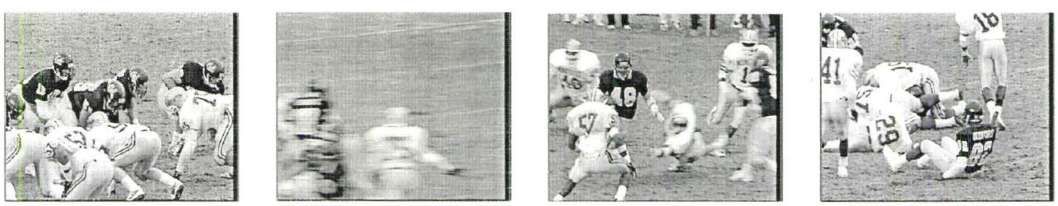
Figure A.2: Sample frame numbers from QCIF format test sequences: (a) **Silent** and (b) **Mother&daughter**.

#000 #50 #100 #150

(a)

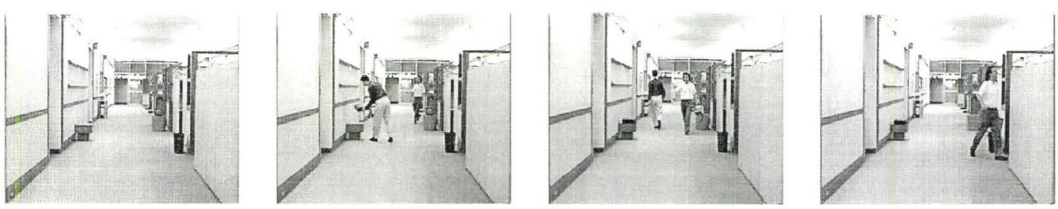
#000 #80 #160 #240

(b)



#000 #80 #160 #240

(c)



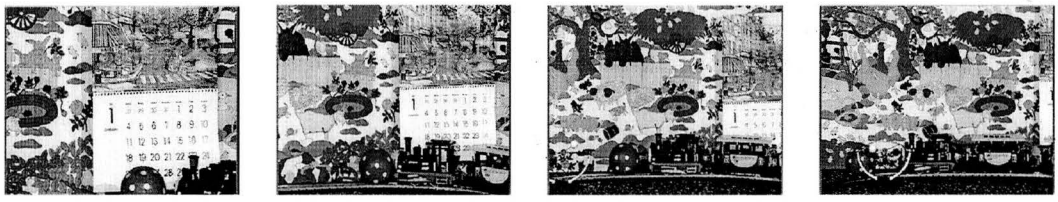
#000 #100 #200 #300

(d)

Figure A.3: Sample frame numbers from CIF format test sequences: (a) **Bus**, (b) **Flower**, (c) **Football** and (d) **Hall**.

#000 #600 #1200 #1800

(a)

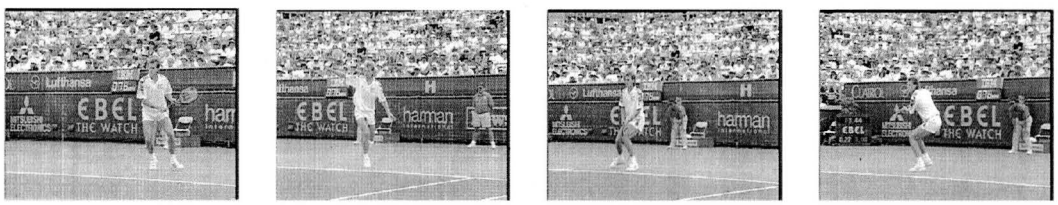


#000 #100 #200 #300

(b)

#000 #100 #200 #300

(c)



#000 #30 #60 #90

(d)

Figure A.4: Sample frame numbers from CIF format test sequences: (a) **Highway**, (b) **Mobile**, (c) **News** and (d) **Stefan**.

Bibliography

[1] T. Wiegand, H. Schwarz, , A. Joch, F. Kossentini, and G. J. Sullivan, "Rate-constrained coder control and comparison of video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 688–703, July 2003.

[2] B. Girod, A. M. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed video coding," *In Proc. IEEE*, vol. 93, no. 1, pp. 71–83, Jan. 2005.

[3] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inform. Theory*, vol. 19, no. 4, pp. 471–480, July 1973.

[4] M. Ghanbari, "The cross-search algorithm for motion estimation," *IEEE Trans. Commun.*, vol. 38, no. 7, pp. 950–953, July 1990.

[5] L.-M. Po and W.-C. Ma, "A novel four-step search algorithm for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 3, pp. 313–317, June 1996.

[6] K. Koga, K. Iinuma, A. Hirano, and T. Ishiguro, "Motion compensated inter-frame coding for video conferencing," in *Proc. Nat. Telecommun. Conference*, New Orleans, LA, Nov. 29 – Dec. 3 1981, vol. 86, pp. G5.3.1–G5.3.5.

[7] R. Li, B. Zeng, and M. L. Liou, "A new three-step search algorithm for block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, no. 4, pp. 438–442, Aug. 1994.

[8] X. Jing and L.-P. Chau, "An efficient three-step search algorithm for block motion estimation," *IEEE Trans. Multimedia*, vol. 6, no. 3, pp. 435–438, June 2004.

[9] L.-K. Liu and E. Feig, "A block-based gradient descent search algorithm for block motion estimation in video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 4, pp. 419–422, Aug. 1996.

[10] Y. Nie and K.-K. Ma, "Adaptive rood pattern search for fast block-matching motion estimation," *IEEE Trans. Image Process.*, vol. 11, no. 12, pp. 1442–1449, Dec. 2002.

[11] K.-K. Ma and G. Qiu, "An inproved adaptive rood pattern search for fast block-matching motion estimation in JVT/H.26L," in *Proc. IEEE Int. Symp. Circuits and Syst.*, May 2003, vol. 2, pp. 708–711.

[12] S. Zhu and K.-K. Ma, "A new diamond search algorithm for fast block matching motion estimation," in *Proc. IEEE Int. Conf. Inform. Commun. Signal Process.*, Singapore, Sept. 1997, pp. 292–296.

[13] C.-H. Cheung and L.-M. Po, "A novel cross-diamond search algorithm for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 12, pp. 1168–1177, Dec. 2002.

[14] C. Zhu, X. Lin, and L.-P. Chau, "Hexagon based search pattern for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 5, pp. 349–355, May 2002.

[15] C.-H. Cheung and L.-M. Po, "Novel cross-diamond-hexagonal search algorithms for fast block motion estimation," *IEEE Trans. Multimedia*, vol. 7, no. 1, pp. 16–22, Feb. 2005.

[16] T.-H. Tsai and Y.-N. Pan, "A novel 3-d predict hexagon search algorithm for fast block motion estimation on H.264 video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 12, pp. 1542–1549, Dec. 2006.

[17] Z. B. Chen, P. Zhou, and Y. He, "Fast integer pel and fractional pel motion estimation for JVT JVT-F017.dog, 6th Meeting, Awaji, Japan, Dec. 5-13, 2002, Dec 2002.

[18] A. Chang, O. C. Au, and Y. M. Yeung, "A novel approach to fast multi-frame selection for H.264 video coding," in *Proc. IEEE Int. Symp. Circuits and Syst.*, May 2003, vol. 2, pp. 704–707.

[19] C.-W. Ting, L.-M. Po, and C.-H. Cheung, "Center-biased frame selection algorithms for fast multi-frame motion estimation in H.264," in *Proc. IEEE Int. Conf. Neural Networks and Signal Process.*, Nanjing, China, Dec. 2003, vol. 2, pp. 1258–1261.

[20] H.-J. Li, C.-T. Hsu, and M.-J. Chen, "Fast multiple reference frame selection method for motion estimation in JVT/H.264," in *The 2004 IEEE Asia-Pacific Conference on Circuits and Syst.*, Dec. 2004, vol. 1, pp. 605–608.

[21] L. Shen, Z. Liu, Z. Zhang, and G. Wang, "An adaptive and fast multiframe selection algorithm for H.264 video coding," *IEEE Signal Process. Lett.*, vol. 14, no. 11, pp. 836–839, Nov. 2007.

[22] X. Li, E. Q. Li, and Y. K. Chen, "Fast multi-frame motion estimation algorithm with adaptive search strategies in H.264," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Quebec, Canada, May 17–21 2004, vol. 3, pp. 369–372.

[23] Y. Su and M.-T. Sun, "Fast multiple reference frame motion estimation for H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 3, pp. 447–452, Mar. 2006.

[24] S.-C. H and Y.-C. H, "Fast multi-frame motion estimation for H.264/AVC system.," *Journal of Signal, Image and Video Processing.*, vol. 3, no. 1, pp. 447–452, Feb. 2009.

[25] B. Jeon and J. Lee, "Fast mode decision for H.264ITU – TQ.6/16 2003.

[26] L. Yang, K. Yu, J. Li, and S. Li, "An effective variable block-size early termination algorithm for H.264 video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 9, pp. 784–788, June 2005.

[27] D. Wu, S. Wu, P. P. Lim, F. Pan, Z. G. Li, and X. Lin, "Block inter mode decision for fast encoding of H.264," in *IEEE Int. Conference on Acoust., Speech, and Signal Process.*, Quebec, Canada, May 2004, vol. 3, pp. 181–184.

[28] S. Spinsante, F. Chiaraluce, E. Gambi, and C. Falasconi, "Mode decision optimization issues in H.264 video coding," in *IEEE Int. Symp. on Signal Process. and Inform. Technol.*, Athens, Greece, Dec. 2005, pp. 624–629.

[29] S. Gao and T. Lu, "Efficient mode decision algorithm in H.264 for mobile video communications," in *IEEE Int. Conference on Commun., Circuits and Syst. Proc.*, Guilin, China, June 2006, vol. 1, pp. 105–108.

[30] I. Choi, J. Lee, and B. Jeon, "Fast coding mode selection with rate-distortion optimization for mpeg-4 part-10 avc/h.264," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 12, pp. 1557 – 1561, Dec 2006.

[31] H. Wang, S. Kwong, and C.-W. Kok, "An efficient mode decision algorithm for H.264/AVC encoding optimization," *IEEE Trans. Multimedia*, vol. 9, no. 4, pp. 882–888, June 2007.

[32] T.-Y. Kuo and C.-H. Chan, "Fast variable block size motion estimation for H.264 using likelihood and correlation of motion field," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 10, pp. 1185–1195, Oct. 2006.

[33] D. Wu, F. Pan, K. P. Lim, S. Wu, Z. G. Li, X. Lin, S. Rahardja, and C. C. Ko, "Fast intermode decision in H.264/AVC video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 6, pp. 953–958, July 2005.

[34] L.-K. Liu, "Dynamic search range motion estimation for video coding," in *IEEE Workshop on Multimeida Signal Process.*, June 1997, vol. 1, pp. 207–212.

[35] S. Goel, Y. Ismail, and M. A. Bayoumi, "Adaptive search window size algorithm for fast motion estimation in H.264/AVC standard," in *The 48th Midwest Symp. on Circuits and Syst.*, Aug. 2005, vol. 2, pp. 1557–1560.

[36] G. Bailo, M. Bariani, I. Barbieri, and M. Raggio, "Search window size decision for motion estimation algorithm in H.264 video coder," in *IEEE Int. Conf. Image process.*, Oct. 2004, vol. 3, pp. 1453–1456.

[37] R. Puri, A. Majumdar, P. Ishwar, and K. Ramchandran, "Distributed video coding in wireless sensor networks," *IEEE Signal Process. Mag.*, vol. 23, no. 4, pp. 94–106, July 2006.

[38] G. Guillemot, F. Pereira, L. Torres, T. Ebrahimi, R. Leonardi, and J. Ostermann, "Distributed monoview and multiview video coding," *IEEE Signal Process. Mag.*, pp. 67–76, Sep. 2007.

[39] A. Aaron, S. Rane, R. Zhang, and B. Girod, "Wyner-Ziv coding for video: applications to compression and error resilience," in *IEEE Data Compression Conf.*, Snowbird, UT, USA, Mar. 25–27 2003, pp. 93–102.

[40] M. Valera and S. Velastin, "Intelligent distributed surveillance systems: A review," in *in IEEE Proc. Vis. Image, Signal Process.*, Apr. 2005, vol. 152, pp. 192–2004.

[41] L. Lu, D. He, and A. Jagmohan, "Side information generation for distributed video coding," in *IEEE Int. Conf. Image Process.*, San Antonio, Texas, USA, Sept. 2007, vol. 2, pp. 13–16.

[42] J. Ascenso, C. Brites, and F. Pereira, "Improving frame interpolation with spatial motion smoothing for pixel domain distributed video coding," in *5th EURASIP Conf. on Speech and Image Process., Multimedia Commun. and Service*, Smolenice, Slovak Republic, June 2005.

[43] Z. Li and E. J. Delp, "Wyner – Ziv video side estimator: conventional motion search methods revisited," in *IEEE Int. Conf. Image Process.*, Genova, Italy, Sept. 2005, vol. 1, pp. 825–828.

[44] I. Park and D. W. Capson, "Dynamic reference frame selection for improved motion estimation time in H.264/AVC," in *Proc. IEEE Southwest Symp. on Image Anal. and Interpretation*, Santa Fe, NM, USA, Mar 2008, pp. 97–100.

[45] I. Park and D. W. Capson, "Improved inter mode decision based on residue in H.264/AVC," in *Proc. IEEE Int. Conf. on Multimedia and Expo*, Hanover, Germany, June 2008, pp. 709–712.

[46] T. Wiegand, G. J. Sullivan, G. Bjntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[47] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, and T. Wedi, "Video coding with H.264/AVC: tools, performance, and complexity," *IEEE Circuits Syst. Mag.*, vol. 4, no. 1, pp. 7–28, Mar. 2004.

[48] D. Marpe, H. Schwarz, and T. Wiegand, "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 674–687, July 2003.

[49] Khalid Sayood, *Introduction to data compression*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2006.

[50] Y. T. Chang and C. L. Wang, "A new fast DCT algorithm and its systolic VLSI implementation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 44, no. 11, pp. 959–962, Nov. 1997.

[51] S. F. Hsiao, Y. H. Hu, T. B. Juang, and C. H. Lee, "Efficient VLSI implementations of fast multiplierless approximated DCT using parameterized hardware modules for silicon intellectual property design," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 52, no. 8, pp. 1568–1579, Aug. 2005.

[52] E. Feig, "A fast scaled DCT algorithm," in *Proc. SPIE Int. Soc. Opt. Eng*, 1990, vol. 1244, pp. 2–13.

[53] E. Feig and S. Winograd, "Fast algorithms for the discrete cosine transform," *IEEE Trans. Signal Processing*, vol. 40, pp. 2174–2193, Sept. 1992.

[54] W. Cham, "Development of integer cosine transforms by the principle of dyadic symmetry," in *Proc. Inst. Elect. Eng., Part I*, Aug 1989, vol. 136, pp. 276–282.

[55] J. Liang and T. D. Tran, "Fast multiplierless approximations of the DCT with the lifting scheme," *IEEE Trans. Signal Processing*, vol. 49, no. 12, pp. 3032–3044, Dec. 2001.

[56] R. Kutka, "Fast computation of DCT by statistic adapted look-up tables," in *Proc. IEEE Int. Conf. on Multimedia and Expo*, Lausanne, Switzerland, Aug. 2002, vol. 1, pp. 781–784.

[57] H. Wang and S. Kwong, "Prediction of zero quantized DCT coefficients in H.264/AVC using Hadamard transformed information," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 4, pp. 510–515, Apr. 2008.

[58] J.M.M. Anderson and G.B. Giannakis, "Image motion estimation algorithms using cumulants," *IEEE Trans. Image Process.*, vol. 4, no. 3, pp. 346–357, Mar. 1995.

[59] C.M. Fan, N.M. Namazi, and P.B. Penafiel, "A new image motion estimation algorithm based on the em technique," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 3, pp. 348–352, Mar. 1996.

[60] L.-G. Chen, W.-T. Chen, Y.-S. Jehng, and T.-D. Chiuch, "An efficient parallel motion estimation algorithm for digital image processing," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 1, no. 4, pp. 378–385, Dec 1991.

[61] M.G. Strintzis and I. Kokkinidis, "Maximum likelihood motion estimation in ultrasound image sequences," *IEEE Signal Process. Lett.*, vol. 4, no. 6, pp. 156–157, June 1997.

[62] J. Weng, P. Cohen, and N. Rebibo, "Motion and structure estimation from stereo image sequences," *IEEE Trans. Robot. Autom.*, vol. 8, no. 3, pp. 362–382, June 1992.

[63] H. Li, P. Roivainen, and R. Forchheimer, "3-d motion estimation in model-based facial image coding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 6, pp. 545–555, June 1993.

[64] K. Rijkse, "H.263: video coding for low-bit-rate communication," *IEEE Commun. Mag.*, vol. 34, no. 12, pp. 42–45, Dec. 1996.

[65] Y.-W. Huang, B.-Y. Hsieh, S.-Y. Chien, S.-Y. Ma, and L.-G. Chen, "Analysis and complexity reduction of multiple reference frames motion estimation in

H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 4, pp. 507–522, Apr. 2006.

[66] S.-E. Kim, J.-K. Han, and J.-G. Kim, "An efficient scheme for motion estimation using multireference frames in H.264/AVC," *IEEE Trans. Multimedia*, vol. 8, no. 3, pp. 457–466, June 2006.

[67] Y. W. Huang, B. Y. Hsieh, T. C. Wang, S. Y. Chient, S. Y. Ma, C. F. Shen, and L. G. Chen, "Analysis and reduction of reference frame for motion estimation in MPEG – 4AVC/JVT/H.264," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Hong Kong, Apr. 2003, vol. 3, pp. 145–148.

[68] P. Wu and C. B. Xiao, "An adaptive fast multiple reference frames selection algorithm for H.264/AVC," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Las Vegas, USA, Mar. 2008, pp. 1017–1020.

[69] N. Ozbek and A. M. Tekalp, "Fast H.264/AVC video encoding with multiple frame references," in *IEEE Int. Conf. Image Process.*, Genoa, Italy, Sept. 2005, vol. 1, pp. 11–14.

[70] J. Lee and B Jeon, "Fast mode decision for H.264," in *Proc. IEEE Int. Conf. on Multimedia and Expo*, Taipei, Taiwan, June 2004, pp. 1131–1134.

[71] L. L. Wang and W. C. Siu, "H.264 fast intra mode selection algorithm based on direction difference measure in the pixel domain," in *IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP).*, Taipei, Taiwan, 2009, pp. 1037–1040.

[72] F. Pan, X. Lin, S. Rahardja, K. P. Lim, Z. G. Li, D. Wu, and S. Wu, "Fast mode decision algorithm for intra prediction in H.264/AVC video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 10, pp. 813–822, July 2005.

[73] S. H. Ri, Y. Vatis, and J. Ostermann, "Fast inter-mode decision in an H.264/AVC encoder using mode and lagrangian cost correlation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 2, pp. 302–306, Feb. 2009.

[74] I. Park and D. W. Capson, "Improved motion estimation time using a combination of dynamic reference frame selection and residue-based mode decision," *Journal of Signal, Image and Video Processing (conditionally accepted)*, 2009.

[75] M.-J. Chen, G.-L. Li, Y.-Y. Chiang, and C.-T. Hsu, "Fast multiframe motion estimation algorithms by motion vector composition for the MPEG-4/AVC/H.264 standard," *IEEE Trans. Multimedia*, vol. 8, no. 3, pp. 478–487, June 2006.

[76] Joint Video Team Software JM13.1 [Online] Available: http://iphoome.hhi.de/suehring/tml.

[77] I. Park and D. W. Capson, "Improved side information generation using dynamic motion estimation for distributed video coding," *IEEE Trans. Circuits Syst. Video Technol. (submitted)*, 2009.

[78] ISO/IEC 11172-2:1993 Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s – Part 2: Video, 1993.

[79] ISO/IEC 13818-2:2000 Information technology – Generic coding of moving pictures and associated audio information: Video, 2000.

[80] ITU-T Recommendation H.261 – Video codec for audiovisual services at $p \times 64$ kbits, 1993.

[81] D. Marpe and T. Wiegand, "The H.264/MPEG4 advanced video coding standard and its applications," *IEEE Commun. Mag.*, pp. 134–143, Aug. 2006.

[82] T. Sikora, "The MPEG-4 video standard verification model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 1, pp. 19–31, Feb. 1997.

[83] A. D. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. Inform. Theory*, vol. 22, no. 1, pp. 1–10, Jan. 1976.

[84] R. Puri and K. Ramchandran, "PRISM: a new robust video coding architecture based on distributed compression principles," in *Allerton Conf. on Communication, Control and Computing*, Allerton, IL, USA, Oct. 2002.

[85] L. Liu, Z. Li, and E. J. Delp, "Efficient and low-complexity surveillance video compression using backward-channel aware Wyner – Ziv video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 4, pp. 453–465, Apr. 2009.

[86] A. Aaron, S. Rane, E. Setton, and B. Girod, "Transform-domain Wyner – Ziv codec for video," in *Proc. SPIE Visual Commun. and Image Process., VCIP-2004*, San Jose, CA, USA, Jan. 2004, pp. 520–528.

[87] A. Aaron, S. Rane, E. Setton, and B. Girod, "Transform domain Wyner – Ziv codec for video," in *SPIE Visual Commun. and Image Process.*, San Jose, CA, USA, Jan. 2004.

[88] A. Aaron, E. Setton, and B. Girod, "Towards practical Wyner – Ziv coding of video," in *IEEE Int. Conf. Image Process.*, Barcelona, Spain, Sept. 2003, vol. 3, pp. 869–872.

[89] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo codes," *IEEE Trans. Commun.*, vol. 44, no. 10, pp. 1261–1271, Oct. 1996.

[90] R. Gallager, *Low-Density Parity-Check codes*, Cambridge, MA: MIT Press, 1963.

[91] J. Garcia-Frias and Y. Zhao, "Data compression of correlated non-binary sources using punctured turbo codes," in *IEEE Data Compression Conf.*, Snowbird, UT, USA, Apr. 2–4 2002, pp. 242–251.

[92] M. Sartipi and F. Fekri, "Distributed source coding using short to moderate length rate-compatible LDPC codes: the entire Slepian – Wolf rate region," *IEEE Trans. Commun.*, vol. 56, no. 3, pp. 400–411, Mar. 2008.

[93] S. S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): design and construction," *IEEE Trans. Inform. Theory*, vol. 49, no. 3, pp. 626–643, Mar. 2003.

[94] M. Dalai, R. Leonardi, and F. Pereira, "Improving turbo codec integration in pixel-domain distributed video coding," in *IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP).*, San Diego, CA, USA, 2008, pp. 1116–1119.

[95] J. Ascenso, C. Brites, and F. Pereira, "Design and performance of a novel low-density parity-check code for distributed video coding," in *IEEE Int. Conf. Image Process.*, San Diego, CA, USA, 2008, pp. 1116–1119.

[96] Y. Wang, J. Jeong, and C. Wu, "An approach to side information estimation for Wyner – Ziv video coding," in *IEEE Int. Congress on Image and Signal Process.*, Sanya, Hainan, China, May 2008, vol. 1, pp. 405–410.

[97] G. Berger, R. Coedeken, and J. Richardson, "Motivation and implementation of a software H.264 real-time CIF encoder for mobile TV broadcast applications," *IEEE Trans. Broadcast.*, vol. 53, no. 2, pp. 584–587, June 2007.