Fast Motion Estimation Algorithms for Block-Based Video Coding Encoders

By

Abdelrahman Abdelazim

A thesis submitted in partial fulfilment for the requirements of the degree of Doctor of Philosophy at the University of Central Lancashire

March 2011

The research presented in this thesis was carried out in the
Applied Digital Signal and Image Processing Research Centre,
School of Computing, Engineering and Physical Sciences,
University of Central Lancashire.

uclan
University of Central Lancashire

Student Declaration

Concurrent registration for two or more academic awards

I declare that while registered as a candidate for the research degree, I have not been a registered candidate or enrolled student for another award of the University or other academic or professional institution

Material submitted for another award

I declare that no material contained in the thesis has been used in any other submission for an academic award and is solely my own work

Collaboration

This work presented in this thesis was carried out at the ADSIP (Applied Digital Signal and Image Processing) Research Centre, University of Central Lancashire. The work described in the thesis is entirely the candidate’s own work.

Signature of Candidate ____

Type of Award Doctor of Philosophy

School School of Computing, Engineering and Physical Sciences

Acknowledgements

Writing this part of the thesis gives me a formal opportunity to thank the people who have supported me and consequently had influence on the accomplishment of this work within three years.

I would like to express my deepest gratitude and respect to my supervisors, for their invaluable guidance, time, patience and encouragement throughout this research work. I am speechless, overwhelmed by the unbelievable fact that I had such a great team of supervisors to work with. Firstly, I would like to thank Prof. Djamel Ait-Boudaoud for his support and guidance, both technical and otherwise over the last three years. I would like thank Dr. Martin Varley, to whom I owe the greatest gratitude for his constant support that goes far beyond those three years. Special thanks to Dr. Stephen Mein, for the many enlightening discussions and excellent suggestions.

In addition, I would like to thank my past and present colleagues at the Centre of Applied Digital Signal and Image Processing (ADSIP) with whom I built lasting friendships, helped me in many ways during the past three years. I want to thank Professor Lik-Kwan Shark for giving me many chances to participate in international conferences.

I am also grateful to Prof. Christos Grecos and Dr. Ming Yuan Yang for their guidance and for first introducing me to the area of video Compression.

I also want to thank my parents, my brothers, my wife and my family for constantly encouraging and supporting me in every possible way through the years.

Abstract

Nowadays the amount of digital video applications is rapidly increasing. The amount of raw video data is very large which makes storing, processing, and transmitting video sequences very complex tasks. Furthermore, whilst the demand for enhanced user experience is growing, the sizes of devices capable of performing video processing operations are getting smaller. This further increases the practical limitations encountered when handling these large amounts of data, and makes research on video compression systems and standards very important.

The H.264 is the latest international video coding standard. It compresses high quality video content at low bitrates for a wide range of applications. It uses state-of-the-art coding tools and provides enhanced coding efficiency to provide higher compression capabilities with high perceptual quality. These capabilities have also contributed to significant increase in complexity when implementing the H.264 in real-time applications. Within video coding, motion estimation is a primary contributor to the gain in compression but is also the most computationally intensive part. The objective of this project is designing and combining a series of novel techniques to overcome those limitations. In this thesis, an investigation and four algorithms are proposed which can be classified along three main streams. In the first stream, an investigation was carried out and two algorithms were designed for optimising the motion estimation process for the H.264/AVC whilst maintaining the same quality and the compression rate as the standard. They are based on exploiting frequency domain motion estimation and on the interpolation effect on the motion estimation process. Firstly, the H.264 recommended interpolation and rate distortion methods were examined when frequency domain motion estimation is employed, this investigation has outlined novel improvements for frequency domain motion estimation adaptation. Secondly, a novel fast frequency domain motion estimation algorithm has been designed, the advantage of this approach over standard algorithms is the significant reduction in the encoding complexity it provides for a variety of video sequences. Finally, a novel fast subpixel motion estimation algorithm has been developed, the algorithm adaptively terminates subpixel motion estimation based on the video properties. In the second and third streams the complexity reduction algorithms are further developed to achieve complexity-scalable control of the standard scalable and multiview extensions where more data and flexibilities are incorporated to enhance the end-user experiences.

The proposed algorithms offer the following developments and contributions. The application of the interpolation effect to reduce the encoding complexity is unique. The developed algorithms are flexible in their applications and can be combined with different fast algorithms. The conducted experiments show significant speed improvements, thus making a novel contribution to the implementation of real-time H.264 standard encoders in computationally constrained environments such as low-power mobile devices and general purpose computers.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AVC	Advanced Video Coding
BMA	Block Matching Algorithms
CABAC	Context-Based Adaptive Binary Arithmetic Coding
CCITT	International Telegraph and Telephone Consultative Committee
CIF	Common International Format
DCT	Discrete Cosine Transform
DPCM	Differential Pulse Code Modulation
DW	Distortion Weight
FSBM	Full Search Block Matching
FR	Full Reference
HD	High Definition
HVS	Human Visual System
IDCT	Inverse Discrete Cosine Transform
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
ITU-T	Telecommunications Union-Telecommunication standardization sector
JM	Joint Model
JPEG	Joint Photographic Experts Group
NR	Non Reference
MAFD	Mean Absolute Frame Difference
MB	Macroblock
MC	Motion Compensation
MCP	Motion-compensated Prediction
ME	Motion Estimation
MD	Mode Decision
MPEG	Motion Picture Experts Group
MSE	Mean-Square Error
MV	Motion Vector
MVC	Multiview Video Coding
MVD	Motion Vector Difference
MVV	MultiView Video
PSNR	Peak Signal to Noise Ratio
QCIF	Quarter CIF
QP	Quantisation Parameter
RD	Rate Distortion
RF	Reference Frame

RR	Reduced Reference
SAD	Sum of Absolute Difference
SATD	Sum of Absolute Transformed Differences
SSD	Sum of the Squared Differences
SVC	Scalable Video Coding
UVLC	Universal Variable Length Coding
VHS	Video Home System

CHAPTER 1
INTRODUCTION

1.1 Motivation

Recent advances in digital technologies have paved the way to the development of numerous real-time applications deemed too complex in the past. A vast array of those applications requires transmission and storage of digital videos. Examples include but are not limited to: digital TV, video streaming, multimedia communications, remote monitoring, videophones and video conferencing. Advances in digital video can be classified as one of the most influential modern technologies; this is due to the fast wide spread use of digital video applications into everyday life. Consequently, over the last three decades, high-quality digital video has been the goal of companies, researchers and standardisation bodies [1].

Amongst the various stages required for transmitting digital video from a source to a destination, the video compression stage has played a significant part in the realisation of most of the digital video technologies. In this stage the size of digital video is compressed for transmission and for storage, and then decompressed for display. As network bandwidths and storage capacities continue to increase, there is an ongoing debate about why video compression is needed and why there is such a significant effort to make it better [1]. Perhaps in the future video compression will not be necessary, but at present, for several reasons, it is more than ever an important field for research and development. First, advances in mobile phones technology have led to the development of a generation of small devices capable of processing videos whilst operating using limited power supply and channel bandwidths. As the amount, demands and quality of video information processed on those devices are increasing, the need for efficient video compression techniques is also increasing. Second, in a typical television application, raw digital video requires an average data rate of 216 Mbps [2] while high-definition television requires an average of 1.5 Gbps. This demonstrates that digital video has huge amounts of data that even with constant advances in storage and transmission capacity, there is an obvious need for effective video compression technologies. Furthermore, emerging new technologies for enhancing the user experience such as 3-D television and free viewpoint video result in an even larger amounts of video data to be processed and transmitted. With the limited bandwidth of today's networks video compression is likely to be an essential component video processing for many years to come. Figure 1.1 illustrates a simple video coding system.

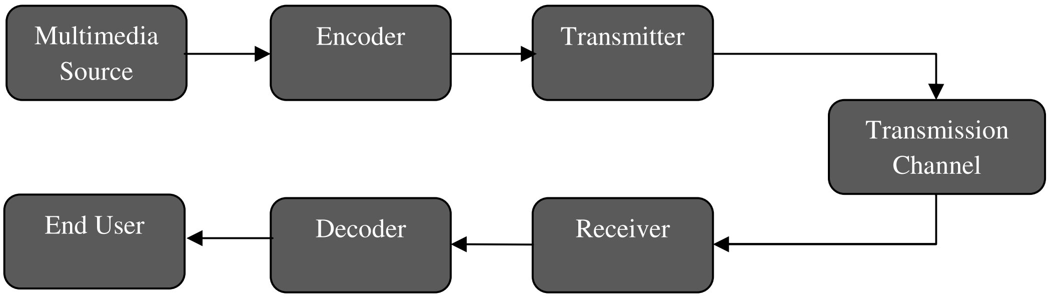


Figure 1.1- Simple information coding system

As shown in the figure, a typical video compression (or coding) system consists of an encoder that converts the video sequence to a compact representation useful for transmission or storage, while the decoder performs the opposite operation. The main idea behind video compression is to remove redundancies from the signals. This is carried out in the spatial domain within individual frames and in the temporal domain between neighbouring frames. Due to the fact video frames are typically displayed at a frame rate of 20 to 30 frames per second to the user, it is easy to understand that neighbouring frames often show high resemblance, hence removing the temporal redundancy can accomplish high compression ratios in practice. This is normally achieved in two steps; first, a Motion Estimation (ME) technique is used to calculate the motion distance, where blocks are defined areas within the frame. Secondly, with the available motion information, the residual between the current encoded frame and the previous frame is compensated and what is called Motion Compensation (MC).

This thesis focused on the development of novel low computational cost algorithms for video coding based on H.264 standards. Through these algorithms, new ideas are explored for potentially improving the standards.

1.2 History of Video Compression

Raw video consists of a sequence of frames or pictures, therefore the principles of video compression are primarily based on image compression. Since the early 1980s the field of image and video coding has seen considerable progress. Many video coding systems and standards have been developed through the years. The most popular standards have been published by the International Telecommunications Union-Telecommunication standardization sector (ITU-T) [3], or by the International Organization for Standardization (ISO) [4] in conjunction with the International Electrotechnical Commission (IEC) [5]. The more recent standards were the result of the collaboration between experts from the two standardisation bodies.

In 1984, a Differential Pulse Code Modulation (DPCM) based video coding standard (H.120) was developed by the International Telegraph and Telephone Consultative Committee (CCITT, forerunner of the ITU-T). The standard worked on line-by-line basis and achieved a target rate of 2 Mbits/s with good spatial resolution but with very poor temporal quality. An improvement of this standard was submitted to the ITU-T in the late 1980s. The new technology was based on the Discrete Cosine Transform (DCT). In parallel to ITU-T's investigation the Joint Photographic Experts Group (JPEG) was also interested in compression of static images, based on DCT as well. Also in the late 1980s, several lossy compression methods began to be used. Examples of this include: run-length encoding, with codewords determined by Huffman coding and lossy DCT, then Huffman or arithmetic coding.

The H.261 [6] standard was recommended in 1988. This standard is considered the first practical digital video coding standard as all subsequent standards are based on the H.261 design. It was the first standard in which 16×16 array of luma samples called MacroBlock (MB), and inter-picture prediction to reduce temporal redundancy emerged. Transform coding using an 8×8 (DCT) to reduce the spatial redundancy was used.

During the 1990s, the Motion Picture Experts Group (MPEG) was aiming to develop a video codec capable of compressing movies onto hard disks, such as CD-ROMs, with a performance comparable to that of Video Home System (VHS) quality. MPEG accomplished this task by developing the MPEG-l [7] standard based on the H.261 framework. In MPEG-1 the speed was traded for compression efficiency since it mainly targeted video storage applications. MPEG-l decoders/players were rapidly adapted to multimedia on computers especially with the release of operating systems or multimedia applications for PC and Mac platforms. In 1994 a new generation of MPEG, called MPEG-2 was developed and soon implemented in several applications. MPEG-2 [8] had great successes in the digital video industry specifically in digital television broadcasting and DVD-Video. A slightly improved version of MPEG-2, called MPEG-3, was to be used for coding of High Definition (HD) TV, but since MPEG-2 could itself achieve this, MPEG-.3 standards were folded into MPEG-2.

The need for better compression tools led to the development of two further standards MPEG-4 [9] visual and H.264 Advanced Video Coding (AVC) [10](also called MPEG-4 part-10) . MPEG-4 visual was designed to provide a flexible visual communications using object-oriented processing. In contrast, the H.264 was aiming to exploit previous technology but in a more efficient, robust and practical way. Its main advantages in comparison to previous standards are the wide variety of applications in which it can be used and its versatile design. This standard has shown significant rate distortion improvements when compared to other standards for video compression. The first version of the standard was released in 2003. The H.264 different profiles has a very broad application range that covers products developed by different companies and broadcast satellite TV in different countries.

It is important to note that in all video codec standards, only the decoders have to comply with proper syntax, software based encoding can add extra flexibility to improve the performance

1.3 Problem Statement

From the above section it can be seen that most video compression systems were designed to achieve very high compression efficiency, this has been accomplished by increasing the complexity of the overall encoding and decoding process. This was motivated by the targeted applications which mainly required efficient storage and transmission. However, with the development of a new range of real-time applications the need for efficient coding and decoding schemes has been highlighted. For example, the H.264 video coding standard has shown significant improvements, but it has also increased the overall encoding complexity due to the very refined ME/MC process. It is well known that ME/MC are the most time-consuming components in the coding pipeline [11], thus numerous fast ME/MC algorithms have been introduced to reduce the complexity of those processes.

This research is particularly aimed at managing the complexity of the encoder, because:

1. An increasing number of low-power handheld devices such as mobile phones and PDAs are capable of capturing video, where the captured video needs to be compressed before storage or transmission.

2. Compression efficiency depends on the coding tools and the decision making process employed by the encoder, which carries a significantly higher computational burden compared to the decoder.

Thus, in this work a complexity management framework is proposed for maximising the perceptual quality of coded video in a real-time processing power-constrained environment.

1.4 Research Objectives

The general research aim is to develop low cost algorithms to effectively manage the computational complexity of an H.264/AVC encoder. These proposed fast algorithms should achieve high speeds of operation, acceptable image quality and can lead to an efficient implementation. These algorithms should enable the encoder to make efficient use of available processing resources whilst providing performance comparable to the H.264 standard. Reducing the implementation gap between performance and complexity is the main aim of this work, this is accomplished by efficiently simplifying some of the complex algorithms to allow existing video coding systems to estimate motion fields more efficiently and robustly. The ME complexity problems are to be tackled from different perspectives, based on controlling the standards robust and sophisticated tools more efficiently.

The specific research objectives can be summarised as:

- The interpolation [12] is one of the main strengths of the standard because it provides more accurate spatial motion representation. However, this increases the overall encoding complexity due to the extra steps required to generate the additional spatial samples. Additionally, the H.264 standard current interpolation method impact on frequency domain ME, in particular, when Hadamard transform is used in the different motion estimation layers, is not thoroughly investigated (more details in chapter 5). One objective of this work is to investigate the interpolation's effect on the frequency domain ME and to identify possible improvements.

- Frequency domain ME approaches such as the phase correlation [13] method have the advantage of being more computationally efficient when compared to the standard block matching technique [10]. Although they provide the same quality as the standard, they result in higher bitrates as they find matches that represents the true motion rather than the rate-distortion optimised matches. An objective of this project is to design an efficient ME scheme that utilises the advantages of both methods. This has been done by adapting a frequency domain ME algorithm (the phase correlation approach) and implementing it in the standard as a pre-processing step in the motion estimation/compensation process.

- The ME process consists of two stages: integer-pixel motion search and fractional-pixel motion search. The fractional-pixel positions are generated by interpolation filters that applied to all the frames, and fractional-pixel motion search is performed for all the macroblocks and their enclosed partitions regardless their inherited motion characteristics. Another objective of this project is to overcome this drawback by defining situations when fractional - pixel ME is ineffective and to design an efficient early termination algorithm to reduce the complexity of fractional-pixel ME.

- The final objective is to establish a relationship between the MBs best matches in full and sub-pixel ME, and its effect on the succeeding encoding steps. These findings are used to reduce the complexity and processing time of the H.264

Scalable Video Coding (SVC) [14] and Multiview Video Coding (MVC) [15] extensions, where ME for MBs is repeated several times in different directions.

The proposed work is inherently distinctive when compared to recently reported research work as outlined in the following chapters, since it will reduce the significant computational cost of the sub-pixel ME, as opposed to full pixel ME which has been widely studied. Furthermore, the project has made original contributions to the body of knowledge in video coding such as the investigation of the interpolation's effect on the frequency domain ME and the development of adaptive fast mode decision algorithms for the scalable and the multiview extensions of the H.264/AVC.

1.5 Research Outline and Contributions

1.5.1 Overview

This research has surveyed various widely used techniques in the H.264/AVC video standards. Those techniques have been investigated, implemented in the standard reference software and their performance has been reported and compared against other techniques. However the core contribution is mainly in the ME which is a vital part of the MC process.

1.5.2 Project outline

In order to achieve the main objective, the research project was divided into the following stages.

Stage 1

1. Investigate and evaluate video coding standards and analyse the rate-distortion performance of an H.264/AVC encoder with different coding parameters and identify the main contributors to the computational complexity of the encoder, and examine the standards and techniques which are used in practical systems for coding video.

2. Investigate and apply existing variable/low complexity algorithms to an H.264/AVC codec and analyse the trade-off between complexity, bitrate and video quality.

Stage 2

3. Develop a new reduced complexity algorithm for an H.264/AVC encoder and evaluate the performance of the algorithm. The new algorithm should reduce the computational complexity with minimal loss in rate-distortion performance.

Stage 3

4. Further develop the reduced complexity algorithm, in order to control the computational complexity of the standard extensions.

1.5.3 Contribution

This work has resulted in a variety of contributions in refereed journals and conferences which are summarised in the following list:

Fast sub-pixel Motion Estimation algorithms

In this section novel fast sub-pixel ME algorithms have been proposed. The algorithms are based on the interpolation effect on the ME and have the advantages of providing a saving of up to 16% of the total encoding time when compared to the standard reference software.

1. Abdelrahman Abdelazim, Ming Yuan Yang, Christos Grecos and Djamel Ait Boudaoud. “Selective application of sub-pixel ME and Hadamard transform in H264 AVC”, SPIE Electronic Imaging Conference, San Jose CA , 18-22 January 2009.

2. Abdelrahman Abdelazim, Ming Yuan Yang and Christos Grecos, “Fast Sub-Pixel Motion Estimation based on the interpolation effect on different block sizes for H264/AVC”, Optical Engineering Letters, ISSN 0091-3286, vol. 48, Issue 3, March 2009.

Frequency Domain Motion Estimation

The original contributions in frequency domain ME can be divided into two parts. The first one is represented by a novel study that analyses the effect of the H.264 interpolation and rate distortion optimisation on the Hadamard transform based ME. The second one is the introduction of a novel fast ME algorithm that is based on a frequency domain ME and takes advantage of both frequency domain and block matching ME. The algorithm saves up to 97% of the encoding time for a number of video sequences.

3. Abdelrahman Abdelazim, Martin Varley and Djamel Ait Boudaoud" Effect Of The Hadamard Transform On Motion Estimation Of Different Layers In Video Coding" , Close Range Image Measurement Techniques, Newcastle upon Tyne, 22-24th June 2010.

4. Abdelrahman Abdelazim, Stephen Mein, Martin Varley, Christos Grecos and Djamel Ait Boudaoud "Phase correlation based adaptive mode decision for the H.264/AVC", SPIE Electronic Imaging Conference, San Francisco CA , 23-27 January 2011.

5. Abdelrahman Abdelazim, Stephen Mein, Martin Varley and Djamel Ait Boudaoud "Fast Mode Decision for the H.264/AVC Based on Frequency Domain Motion Estimation", Optical Engineering Letters, ISSN 0091-3286, vol. 50, Issue 7, July 2011.

Scalable and Multiview Video Coding

The original contribution has been extended to include optimising the ME process in the different layers of the SVC optimising the references frame selection in the MVC.

6. Abdelrahman Abdelazim, Stephen Mein, Martin Varley and Djamel-Ait Boudaoud "Low Complexity Hierarchical Prediction Algorithm for H.264/SVC" , The Fourth Pacific-Rim Symposium on Image and Video Technology, Singapore , 14-17 November 2010.

7. Abdelrahman Abdelazim, Stephen Mein, Martin Varley, Christos Grecos and Djamel Ait-Boudaoud "Fast multilayered prediction algorithm for group of pictures in H.264/SVC", SPIE Electronic Imaging Conference, San Francisco CA , 23-27 January 2011.

8. Abdelrahman Abdelazim, Guang Yao Zhang, Stephen Mein, Martin Varley and Djamel Ait-Boudaoud "Fast motion prediction algorithm for multiview video coding", SPIE Defense, Security, and Sensing Conference, Orlando, Florida, 25-29 April 2011.

1.5.4 Organisation of the Thesis

Chapter 2 – Provides an overview of key concepts and fundamental terms used in video compression. Moreover the structure and the methodologies used in the H.264 standard standards are also presented and explained.

Chapter 3 – Explains the different available ways to measure the performance of video compression, then outlines the experimental methods used in this work.

Chapter 4 – Describes a new complexity reduction algorithm for H.264/AVC which uses frequency domain ME. Initially a literature survey was carried out outlining different fast compression algorithms, then the phase correlation technique is explained and utilised to reduce the ME complexity.

Chapter 5 – Discusses the effect of the interpolation on the frequency domain motion estimation.

Chapter 6 – Proposes the implementation of a novel sub-pixel ME search algorithm. The algorithm is simulated and tested and comparisons are made with recently developed algorithms and the standard methods.

Chapter 7 – Describes complexity reduction achieved for the H.264 /SVC by extending the algorithm described in chapter 6.

Chapter 8 – Introduces a low complexity algorithm for the H.264 /MVC. Firstly, the MVC concepts and tools are discussed, then an extension of the algorithms proposed in chapter 6 and 7 is effectively adapted and applied to the MVC.

Chapter 9 – Presents a summary of the main algorithms and a critical review of the results. The advantages and disadvantages of proposed methods are discussed. Ideas for further investigation are also presented. Finally, the thesis concludes by emphasising the relevance of this work to the research problem and the original contributions made.

CHAPTER 2
AN OVERVIEW OF VIDEO CODING

2.1 Introduction

This chapter lays down the necessary ground work for following chapters by providing a brief introduction to the background of video data compression. This is based around the latest video coding standards the H.264.

Although a detailed description of the standards is beyond the scope of this PhD thesis, this chapter provides some essential background information on video coding. It starts by briefly describing the overall process of a basic video coding scheme. Then digital video concepts related to block based video coding are explained. Finally an overview of the H.264 standard is given with a brief explanation of its structure. The ME and MC tools which are the main focus of work described in this thesis contribution are explained in greater details.

2.2 Video and Image Compression

2.2.1 Overview

Compression in general involves removing redundancies from the video sequence. It can be divided into two main types; lossless compression and lossy compression. If the process entails no loss in information, this type of compression is termed lossless compression. In contrast, using lossy compression the sequence can only be recovered partially. Although lossy compression reduces the size at the expense of quality most practical video compression techniques fall into this category. This is due to the limited amount of compression of image and video signals that lossless compression can achieve. Lossy compression schemes can achieve up to 95% higher compression rate than lossless compression schemes, this can be realised when comparing different commercial lossy and lossless encoders. Psychovisual redundancy can also be exploited because of the nature of Human Visual System (HVS) [16]. The HVS does not perceive certain details in pictures. These pictures' details can be discarded to reduce the size further without introducing perceivable errors in the reconstructed picture.

2.2.2 Concepts and Definitions

In an analogue system a video camera produces an analogue signal of an image scanned from left to right and from top to bottom making up a frame [17]. The choice of number of scanned lines per picture is a trade-off between the bandwidth, flicker and resolution. Any video consists of frames of a scene taken at various subsequent intervals in time. Each frame represents the distribution of light energy and wavelength over a finite size area and is expected to be seen by a human viewer [18].

Digital frames have a fixed number of rows and columns of digital values, called picture elements or pixels. These elements hold quantised values that represent the brightness of a given colour at any specific point [19]. The number of bits per pixel is known as pixel depth or bits per pixel.

Frame rate

The number of frames displayed per second. The illusion of motion can be experienced at frame rates as low as 12 frames per second [20]. Standard-Definition television typically uses 24 frames per second, and HD videos can use up to 60 frames per second.

Frame dimensions

The width and height of the image expressed in the number of pixels. Some common formats include:

- **CIF** (Common International Format), defines a video sequence with a resolution of 352 × 288.
- **QCIF** (Quarter CIF), defines a video sequence with a resolution of 176 × 144.

Colour Spaces

Visual information at each sample point may be represented by the values of three basic colour components Red (R), Green (G) and Blue (B). This is called the RGB colour space. Each value is stored in an ‘n’ bit number. For example, an 8-bit number can store 256 levels to represent each colour component.

The YCrCb colour space is widely used to represent digital video. The luminance component 'Y' is extracted using a weighted average of the three colour components R, G and B. The components Cr and Cb are called the chrominance (or colour difference) components. Cr is the red chrominance component (Cr = R - Y) and the blue chrominance component is Cb, where Cb = B - Y. The HVS has less sensitivity to colour information than luminance (light intensity) information [21]. Therefore, with the separation of luminance information from the colour information, it is possible to represent colour information with a lower resolution than the luminance information. The YCbCr colour space and its variations also referred to as YUV. The typical main component of YUV colour signal consists of a luminance signal Y representing brightness and two chrominance signal U and V representing colour.

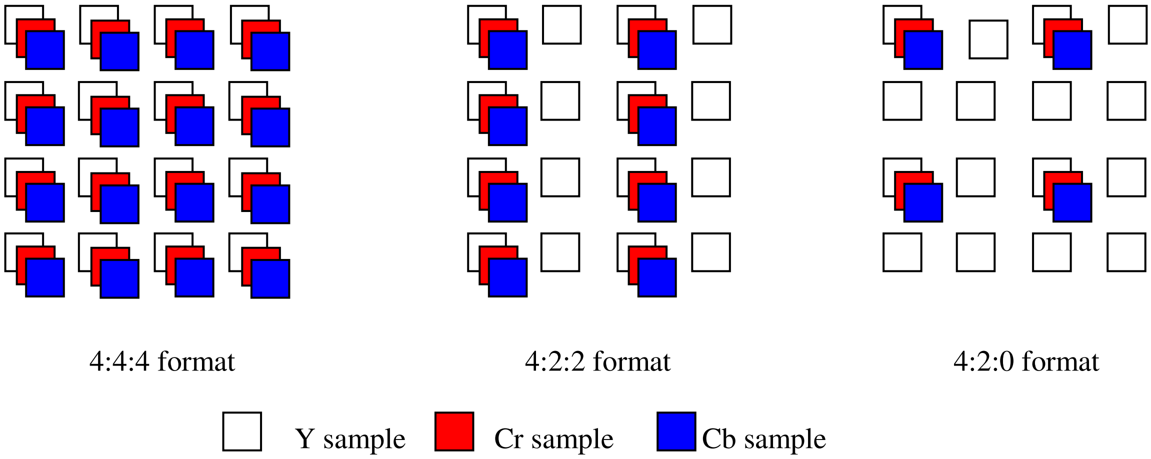


Figure 2.1 - Sub-sampling patterns for chrominance components

In 4:4:4 format, each pixel position has both luminance and chrominance (luma and chroma) samples at full resolution. In 4:2:2 format, chroma components are sub-sampled (every other pixel) in horizontal direction. In 4:2:0 format, chroma samples are sub-sampled in both vertical and horizontal directions. This is the most popular format used in entertainment quality applications such as DVD video. Figure 2.1 shows the different YUV formats.

Macroblock

An image is made up of a rectangular array of pixel values. These pixel values are grouped into blocks. Most existing systems use blocks of a regular size, such as 8×8 or 16×16 pixels [8, 10]. A larger block size can lead to more efficient coding but requires more computational power.

2.3 General Encoder-Decoder System Overview

2.3.1 Hybrid Video Coding

Standard video compression techniques [22-26] are often referred to as hybrid techniques because they make use of several compression tools simultaneously.

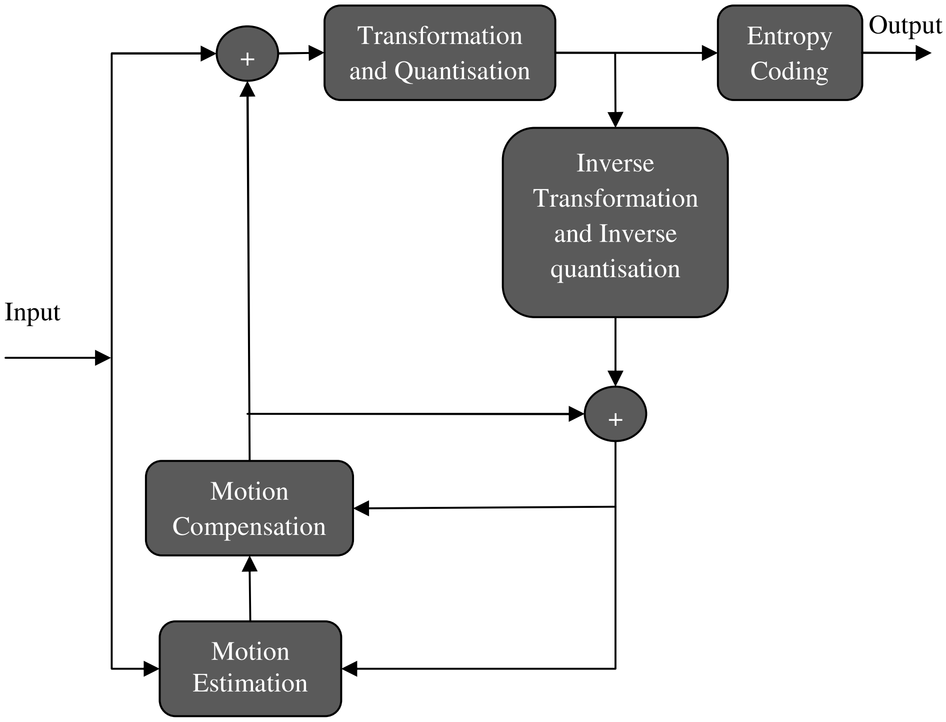

+
Transformation and Quantisation
Entropy Coding
Output
Inverse Transformation and Inverse quantisation
Input
+
Motion Compensation
Motion Estimation

Figure 2.2- Block diagram of the hybrid video encoding system

Each such tool can be used independently, or in conjunction with the other methods. The methods common to all standards video encoding systems are colour sub-sampling, MC, frequency transform, quantisation, and lossless or entropy encoding. Figure 2.2 shows the structure and interconnections of the hybrid coding scheme used by H.264 and other video compression standards.

To help explain the process, assume that there is a simple video sequence that consists of two frames as shown in figure 2.3. At the beginning of the encoding process the first frame is processed in units of a MB, following the path shown in figure 2.2.

Figure 2.3- Two frames from bus test video sequence

2.3.2 Transform Coding

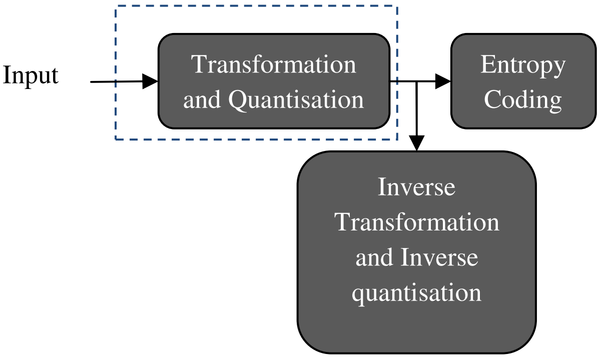

Input
Transformation and Quantisation
Entropy Coding
Inverse Transformation and Inverse quantisation

Figure 2.4 - Transformation and quantisation

At the beginning the first frame goes through the transformation and quantisation stage. The transform stage converts the frame data into another domain; the transform domain. The purpose of this is to distribute the data more favourably for compression and to concentrate most of the energy into a small number of values. This is due to the fact that the image energy of most natural scenes is mainly concentrated in the low frequency region, and hence into a few transform coefficient [1]. As a result, discarding insignificant coefficients has a minimum impact on the perceived reconstructed video quality. Most transform techniques operate on blocks of samples, therefore a frames are processed in units of a block. The transform of a block of samples is given by:

$$Y = HXH^T \tag{2.1}$$

where X is a matrix of samples and Y is the resulting matrix of coefficients and H is the transform matrix.

Whilst most prior standards exploit the DCT, the H.264 applies a separable integer transform with similar properties as DCT to avoid inverse-transform mismatches. Figure 2.5-(a) shows the 4×4 DCT matrix and figure 2.5-(b) illustrates the 4×4 integer transform matrix.

$$H = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.653 & 0.271 & 0.271 & -0.653 \\ 0.5 & -0.5 & -0.5 & 0.5 \\ 0.271 & -0.653 & -0.653 & 0.271 \end{bmatrix} \qquad H = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}$$

(a) **(b)**

Figure 2.5 - (a) 4×4 DCT matrix - (b) 4×4 integer transform matrix

If a 4×4 block is selected from a frame as shown in figure 2.6-(a), applying a transform will result in producing transform coefficients as shown in 2.6-(b).

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \\ x_{41} & x_{42} & x_{43} & x_{44} \end{bmatrix} \qquad Y = \begin{bmatrix} DC & ac_{12} & ac_{13} & ac_{14} \\ ac_{21} & ac_{22} & ac_{23} & ac_{24} \\ ac_{31} & ac_{32} & ac_{33} & ac_{34} \\ ac_{41} & ac_{42} & ac_{43} & ac_{44} \end{bmatrix}$$

(a) **(b)**

Figure 2.6- (a) a 4×4 block is selected from a frame - (b) 4×4 transform coefficients

The resulting matrix always contains an entry with a very large magnitude in comparison to the other coefficients. This entry is located in the top-left corner and is referred to as the DC coefficient, while the remaining coefficients are called the AC coefficients**.**

2.3.3 Quantisation

The quantisation step reduces the range of the values that can be used to represent a sample, hence expressing the signal with a fewer number of bits. An example of a quantisation process is of rounding a fraction number to integer. The Quantisation Parameter (QP) is the step size between the new range values. Thus, large QP means high compression and reduced quality and small QP reflects low compression and high similarity between the original and compressed frames. The encoder quantiser is designed to eliminate relatively insignificant coefficient values and reduce the number of bits for significant coefficient. In H.264 the QP can take 52 values, arranged so that an increase of 1 means an increase of quantisation step size by approximately 12%. This also means a reduction of bitrate by approximately 12% [10]. The quantisation process is inherently lossy.

After the transformation step, since very little of the energy of the block will be contained in the AC coefficients, these coefficients can be quantised heavily. Most AC coefficients are quantised to zero with little impact on the quality of the encoded video sequences.

Following the transformation and quantisation step the quantised transform coefficients of the current frame get passed to the entropy coding stage and the inverse transformation and inverse quantisation stage.

2.3.4 Entropy Coding

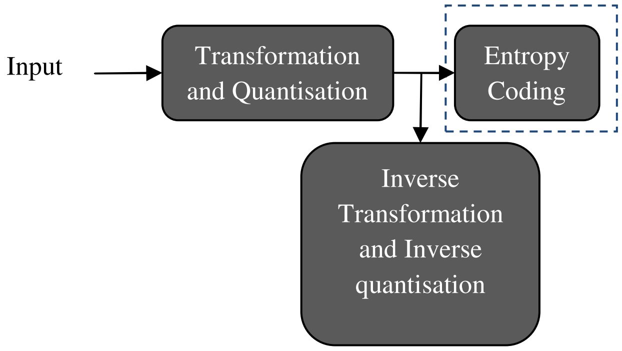

Input
Transformation and Quantisation
Entropy Coding
Inverse Transformation and Inverse quantisation

Figure 2.7 - Entropy Coding

In this step, series of symbols representing elements of the video sequence is converted into a compressed bitstream suitable for transmission or storage. It works by grouping similar frequencies together starting from the DC coefficient and following a zigzag path as illustrated in figure 2.8. The coefficients are copied into a one-dimensional array in that order.

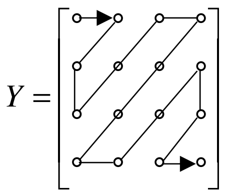


Figure 2.8 - Entropy coding zigzag path

Since the nonzero DCT coefficients are clustered around the top-left (DC) copying the information using this approach results in an array that have the non-zero coefficients at the start followed by a long sequence of zeros. Following this, one of the widely used coding schemes is used; variable-length coding, Huffman coding, arithmetic coding, run-length coding or a combination of them. They all share the fundamental idea of using fewer number of bits for more frequent symbols and greater number of bits for less frequent symbols.

The H.264 standard supports two entropy coding methods: Universal Variable Length Coding (UVLC) and Context-Based Adaptive Binary Arithmetic Coding (CABAC). UVLC [27] employs a fixed codebook which is based on prior probability models for each symbol. In CABAC [28] the probability model is adapted for each symbol according to the context of the symbol and the frequency of occurrence in the previously encoded bitstream. CABAC is a more complex implementation than UVLC, but it has been shown to provide a coding efficiency gain of 9-27% [28].

2.3.4 Inverse Transformation and Inverse Quantisation

As well as encoding and transmitting a frame, the encoder decodes (reconstructs) the frame to provide a reference for further predictions. In these steps the transformation and the quantisation steps are reversed to reconstruct approximation to the original data. Since some of the information is lost during the initial quantisation process the recovered block will not be identical to the original block. However, the retrieved information will be identical to the decoder output as it follows the same inversion process. The transform of a block of samples is given by:

$$X = H^T Y H \qquad (2.2)$$

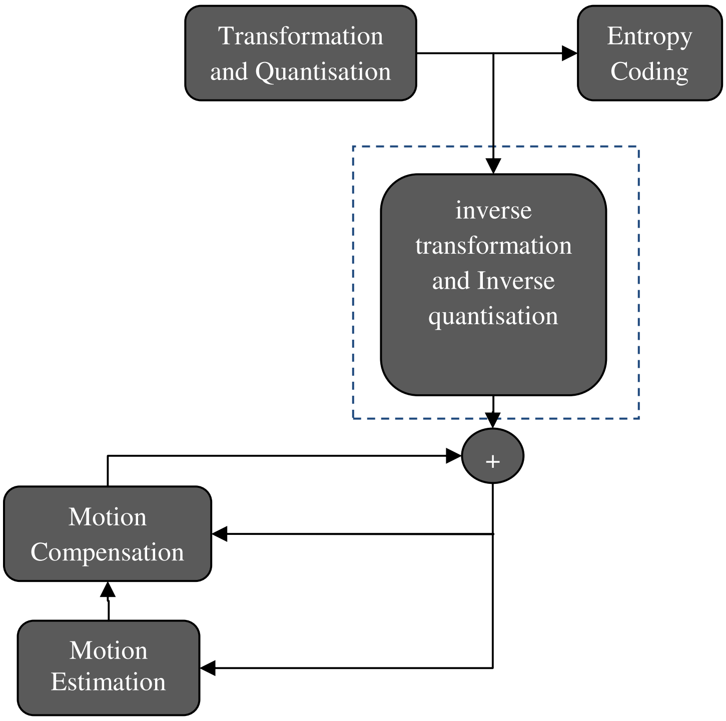

Transformation and Quantisation
Entropy Coding
inverse transformation and Inverse quantisation
+
Motion Compensation
Motion Estimation

Figure 2.9 - Inverse transformation and inverse quantisation

2.3.5 Motion Estimation

This process involves the estimation of the displacement between consecutive frames to exploit the spatial and temporal redundancies inherent in all video sequences.

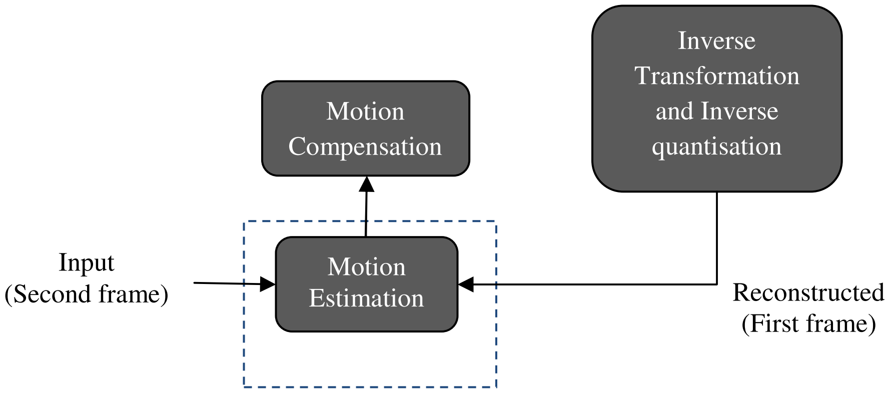

Inverse Transformation and Inverse quantisation
Motion Compensation
Input (Second frame)
Motion Estimation
Reconstructed (First frame)

Figure 2.10 - Motion Estimation

In this stage the current frame raw data and the reconstructed previous frame are given as inputs. Then the Motion Vector (MV) for each block in the current frame is then estimated by finding the block that is closest to the vector in the previous frame within a given search window and according to a matching criterion. Typically, a cost function is employed to measure the mismatch between the block and the candidate blocks. A flexibly defined search window effectively dictates the maximum allowable inter-frame displacement, reducing the number of match positions that can be evaluated.

2.3.6 Motion Compensation

In this stage, after the ME stage finds a best match for each block the difference between the best match and the actual values of each block is obtained and encoded.

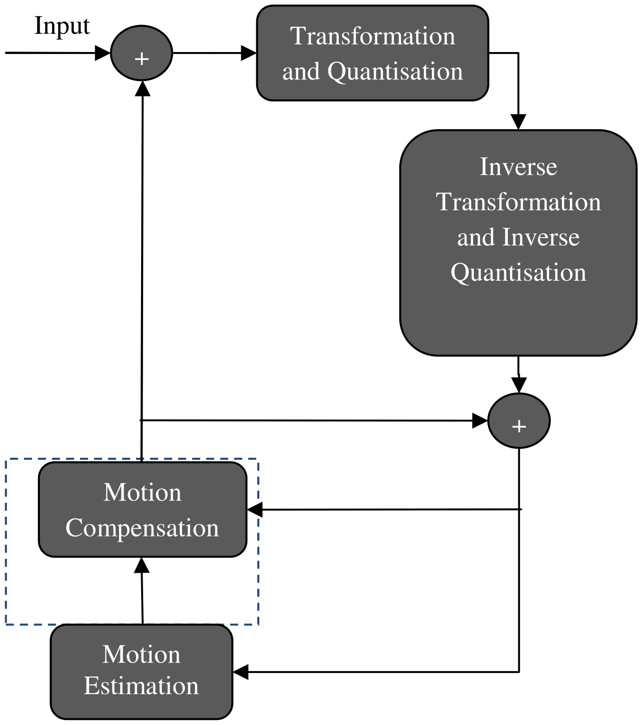

Input
+
Transformation and Quantisation
Inverse Transformation and Inverse Quantisation
+
Motion Compensation
Motion Estimation

Figure 2.11 - Motion Compensation

2.4 H.264/AVC Standard

2.4.1 Overview

In common with earlier coding standards, H.264 only defines the syntax of an encoded video bitstream together with the method of decoding this bitstream. This allows flexibility for individuals to implement and design their optimised version of the encoder as proposed in this work.

The input video signal is divided into frames each consisting of a number of MBs, each of which contains 16×16 luma samples and associated chroma samples, 8×8 Cb and 8×8 Cr values. Most of the basic functional elements are present in previous standards but the important changes in H.264 occur in the details of each functional block [1].

The new H.264/AVC features include variable block size and quarter sample accurate MC with motion vectors even outside picture boundaries, multiple reference frames selection, decoupling of referencing from display order for flexibility [10]. Also the standard removes the extra delay associated with bi-predictive coding and allows bi-predictive pictures to be used as references for better MC. This section describes the ME and the MC process in the standard in great details then outlines the computational complexity sources in the H.264.

2.4.2 Motion Estimation in H.264/AVC

Block Matching Algorithms (BMA)

BMAs are popular methods for ME because of their simplicity and ease of implementation. If the frames are divided into non-overlapping blocks and each block is compared with its counterpart in the previous frame, in order to find an area that is similar. The similar area in the Reference Frame (RF) is known as a best-match. The relative difference in locations is the MV. Figure 2.12 illustrates the block matching method.

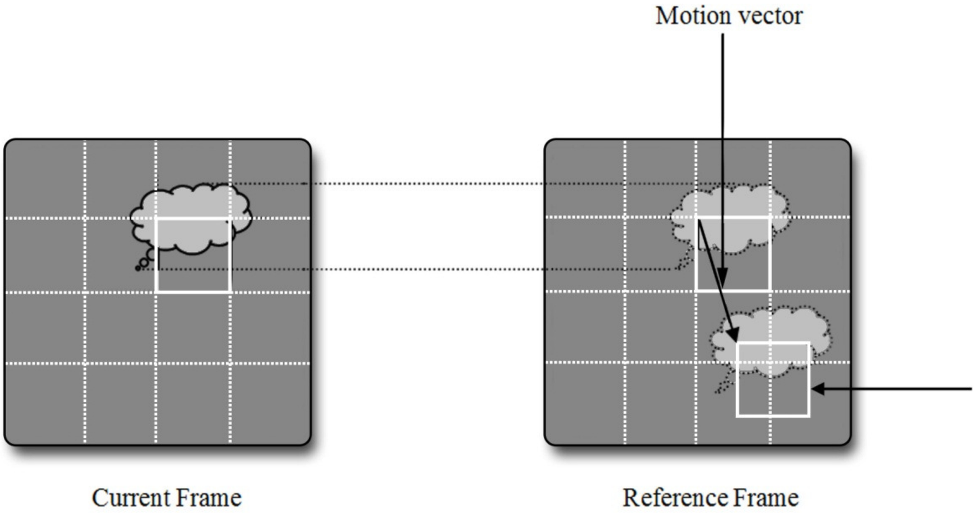


Figure 2.12 - Block matching algorithm

Full Search Block-matching

This is the most common ME approach [29] where a subset of positions is chosen to find the best match. The positions are defined by a rectangular window in the RF, typically for most applications a search ranges from 8 to 64 pixels in each dimension is used in the H.264 standard reference software.

Full-Pixel ME

In the first stage of the ME, an integer-pixel motion search is performed for each square block of the frame to be encoded in order to find one (or more) displacement vector(s) within a search range. The best match is the position that minimises the Lagrangian cost function J_{motion} [30]:

$$J_{motion} = D_{motion} + \lambda_{motion} R_{motion} \tag{2.3}$$

where λ_{motion} is a Lagrangian multiplier, D_{motion} is an error measure between the candidate MB taken from the reference frame(s) and the current MB, R_{motion} stands for the number of bits required to encode the difference between the motion vector(s) and its prediction from the neighbouring MBs (differential coding).

The most common error measures are the Sum of Absolute Difference (SAD) and the Sum of Absolute Transformed Differences ($SATD$). In particular, for any given block of pixels, the SAD between the current MB and the reference candidate MB is computed using the following equation [30]

$$SAD = \sum_{ij} | C_{ij} - R_{ij} | \tag{2.4}$$

Where C_{ij} is a pixel from the current MB and R_{ij} is corresponding pixel of the reference candidate MB.

The Lagrangian cost can also be minimised in the frequency domain, in a very similar manner to the pixel domain. As mentioned above, $SATD$ can be used in equation (2.3) instead of SAD. Central to the calculation of $SATD$ is the 4×4 Hadamard transform which is an alternative to the 4×4 DCT transform. The transform matrix H used is shown in figure 2.13 below (not normalised):

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

Figure 2.13 - Hadamard transform matrix

Since H is a symmetric matrix, it is equal to its own transpose. By using this matrix the $SATD$ is computed using equation (2.5) below:

$$SATD = \frac{\sum_{i,j} | H \times (C_{ij} - R_{ij}) \times H |}{2} \tag{2.5}$$

Where C_{ij} and R_{ij} are the same as in equation (2.4) and H is the matrix in figure 2.13.

Sub-pixel ME

After the integer-pixel motion search finds the best match, the values at half-pixel positions around the best match are interpolated by applying a one-dimensional 6-tap FIR filter horizontally and vertically. Then the values of the quarter-pixel positions are generated by averaging pixels at integer and half-pixel positions. Figure 2.14 illustrates the interpolated fractional pixel positions. Upper-case letters indicate pixels on the full-pixel grid, while numeric pixels indicate pixels at half-pixel positions and lower case letters indicate pixels in between at quarter-pixel positions [10] and [12].

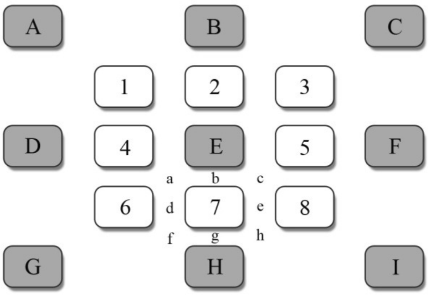


Figure 2.14 - Fractional pixel search positions

For example, in figure 2.14 if the integer best match is position E, the half-pixel positions 1, 2, 3, 4, 5, 6, 7, 8 are searched using equation (2.3). Suppose position 7 is the best match of the half-pixel search. Then the quarter-pixel positions a, b, c, d, e, f, g, h are searched using again equation (2.3).

The application of the Hadamard transform is optional in any resolution and can be enabled or disabled in the configuration files of the encoder. However, for the best Rate Distortion (RD) performance the Joint Model (JM) reference software [31] by default uses *SAD* as the error measure for the integer-pixel motion search, and *SATD* as the error measure for the fractional-pixel motion search.

2.4.3 Motion Compensation in H.264/AVC

Motion compensation in the H.264 involves both intra frame and inter frame coding. The intra frame MC is a mean of exploiting spatial redundancies and inter frame compression is used for exploiting temporal redundancies.

Intra frame MC

In video frames there is a high spatial correlation between neighbouring pixels. Therefore intra frame MC was designed to find good matches for a block of pixels within the same frame. H.264 defines two intra prediction modes, 16×16 and 4×4, the prediction for both modes are based on the neighbouring pixels above and to the left of the current MB.

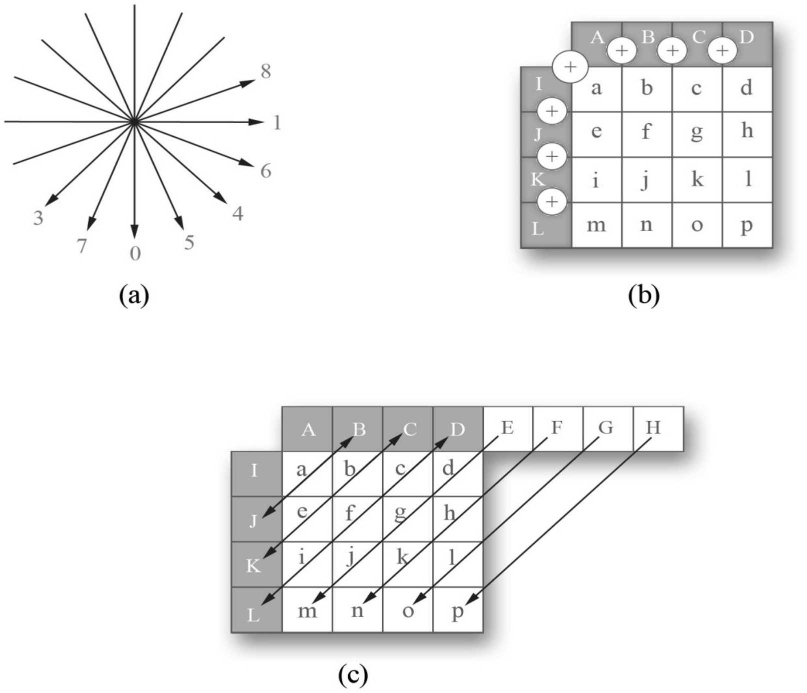


Figure 2.15 - (a) Eight “prediction directions” for Intra_4×4 prediction. - (b) Mode 2-DC - (c) Mode 3- Diagonal Down/left

The 16 samples of the 4×4 block which are labelled as a-p are predicted using prior decoded samples in adjacent blocks labelled as A-L. In addition to the eight modes shown in figure 2.15-(a) a DC prediction mode is included where one value is used to predict the entire 4×4 block as shown in figure 2.15-(b). This value is obtained by averaging the adjacent samples. From figure 2.15-(a) it can be seen that the different prediction modes reflect the possible prediction directions. For example if mode 3 (diagonal-down-left prediction mode) is considered as shown in figure 2.15-(c);

- For p : H is used for prediction
- For g, d, j and m : E is used for prediction
- For e and b : The average of B and K is used for prediction.

For the 16×16 intra mode four prediction modes are supported: prediction modes 0, 1,2 and 4.

Inter frame MC

This compression technique was designed for sequences of video frames, rather than single frames or images. It exploits similarities between successive frames to reduce the amount of data required for storage or transmission. In the H.264 inter frame prediction is limited to translational motion of blocks; for each block, a similar block is identified in a previously encoded frame, and is used for prediction. The difference between the position of the predictive block of pixels in the previous frame and the position of the original pixel block can be represented by a MV with horizontal and vertical components.

Bidirectional MC uses matching blocks from both past frames and future frames to code the current frame.

Unlike other video coding standards, H.264 has many different inter MBs sizes (mode) choices to code a MB as shown in figure 2.16. The modes are related to the partitioning possibilities of a 16×16 luma MB as designated in the following list: (partition width × partition height)

- Inter-mode 1: 16×16 (one partition)
- Inter-mode 2: 16×8 (two partitions)
- Inter-mode 3: 8×16 (two partitions)
- Inter-mode 4: 8×8 (four partitions)
- Inter-mode 5: 8×4 (an 8×8 partition with two sub-partitions)
- Inter-mode 6: 4×8 (an 8×8 partition with two sub-partitions)
- Inter-mode 7: 4×4 (an 8×8 partition with four sub-partitions)

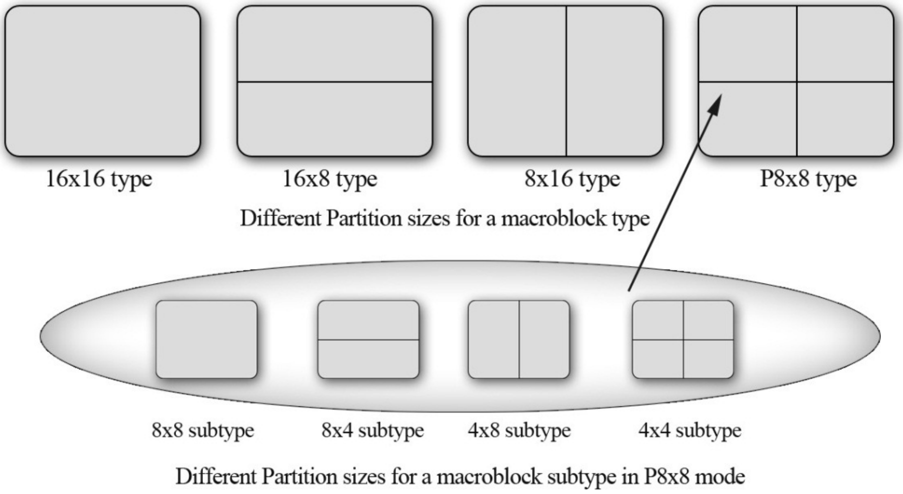

16x16 type
16x8 type
8x16 type
P8x8 type
Different Partition sizes for a macroblock type
8x8 subtype
8x4 subtype
4x8 subtype
4x4 subtype
Different Partition sizes for a macroblock subtype in P8x8 mode

Figure 2.16 - Different partition sizes in a MB

When mode 4 is considered, then modes 5, 6, and 7 must be considered for each of the four individual 8×8 sub- MBs. To choose the best mode, the Lagrangian multiplier is used to compute the cost for each mode and decide on the mode that gives the smallest cost.

Detailed information about the mode selection process in the standard reference software is provided in Appendix A.

Although the exploitation of different MB sizes significantly improves the RD performance, it increases the complexity of the ME process. This is due to the fact that for every MB partition, the integer and fractional motion estimation/compensation have to be performed before deciding on the best mode.

2.4.4 Video Sequence Structure

Intra Frames

I-frames are encoded using only intra methods pixel prediction. Since all predictive pixels are from the same frame, they are coded independently of all other frames.

Predictive Frames

Predictive frames (or P-frames) use intra frame prediction methods as well as intra-frame methods. For P-frame MC, only forward prediction is supported, frames used for prediction must temporally precede the encoded frame. In H.264, multiple RF prediction is permitted, i.e. P-frames pixel blocks may be predicted from any preceding I-frame or P-frame. This feature is useful for encoding transitionally covered background and periodic non-translational motion [32, 33].

Bi-Predictive Frames

Bi-predictive frames (or B-frames) use an expanded set of inter-prediction methods compared to P-frames. Specifically, B-frames support forward and backward prediction for MC, reference frames may occur before or after the encoded frame in the display order of the video sequence. In addition, H.264 B-frames support bi-predictive block compensation [34].

2.4.5 Layer Structure

The standard was designed to contain two layers

1. The Video Coding Layer (VCL) carries the video coded bits.

2. The Network Abstraction Layer (NAL) handles the transportation of VCL data and other header information by encapsulate them in NAL units.

The standard is divided into two layers to ensure efficiency in representing video contents and flexibility to the adaption to a variety of delivery frameworks.

2.4.6 Profiles and Levels

The H.264/AVC standard specifies many profiles to support different applications by providing different tool and capabilities. The following are four used frequently profiles:

1. **Baseline** : designed for low-latency, low-complexity applications.

2. **Main** : designed for applications that require high compression efficiency.

3. **Extended** : designed for enhanced error resilience applications as an extension to the baseline.

4. **High Profiles** : designed for high end consumer use and other applications using high resolution video [35, 36].

2.4.7 Computational Complexity in H.264/AVC

The improvements in the RD performance come with significant complexity increases. These new features not only increase the complexity of H.264/AVC encoders but also of the corresponding decoders, thus impacting adversely in terms of real-time aspects. There are five main sources of computational complexity increases in H.264/AVC namely variable block size motion estimation and compensation; Hadamard transform; RDO mode decision; displacement vector resolution and multiple reference frames.

In the case of variable block size ME and MC, the complexity increase comes from performing ME/MC more than once in order to find the block size with the best rate distortion performance. In the application of Hadamard transform in sub-pixel ME/MC the complexity increase comes from the many arithmetic operations involved. The displacement vector resolution also increases complexity since the optimal MV is found by ME/MC in potentially more than one higher resolution with respect to the original. The use of multiple reference frames increases the complexity proportionally to their number. In [11] a complete analysis of the complexity increase in the standard is presented and compared to previous standards. Reducing this complexity without degrading the RD performance to enable the applicability of the standard in a wide range of applications has become the aim of many companies and researchers.

To clarify the complexity of H.264/AVC, an experiment of complexity analysis is performed here. In this experiment, a video sequence of 100 frames was encoded and the complexity proportions of different encoding modules were recorded as illustrated in figure 2.17.

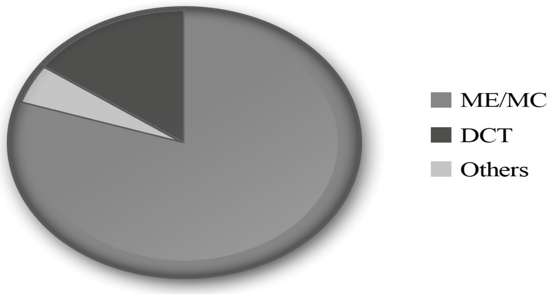


Figure 2.17 - Complexity proportion of different encoding modules in H.264/AVC encoder

According to figure 2.17, the most time-consuming modules of the H.264/AVC encoder are ME and MC which include the interpolation, SAD and SATD calculations. Thus fast ME and mode prediction algorithms are the most effective way to reduce the complexity of video encoders. It is important to note that most video sequences have similar complexity proportion of different encoding modules, particularly when full search ME is employed.

2.5 Summary

In this chapter, the fundamental technology for video coding and the most well known video coding standards are discussed. The video coding tools described in this chapter include motion compensated prediction, transform coding, quantisation and entropy coding form the basis of the reliable and effective coding model that has dominated the field of video compression. The most computationally intensive element of a video encoder is the ME process, requiring up to 80% of the computational resources of typical implementations.

The following chapter explains measurements techniques for computational complexity, bit rate and video quality. Furthermore, the experimental setup used in this thesis is described in details. The later chapters of this thesis outline techniques developed to adaptively control computational complexity while achieving the maximum video quality.

CHAPTER 3
EXPERIMENTS METHOD

3.1 Introduction

This chapter presents the experimental methods used in this research project. There are three main characteristics for the measurement to assess the performance of video compression algorithms. These are compression ratio, image quality and compression speed.

- **Compression ratio** is the measurement of the capability of the storage or data reduction. A higher compression ratio means better data reduction can be achieved.

- **Image quality** is a core measurement which aims to compare the decompressed data to the original data.

- **Compression speed** refers to the computational effort required by the encoding and decoding processes.

These characteristics are usually used for judging the performance of the compression technique. The use of these characteristic measurements depends on the application and use of images for particular requirements. In addition, these characteristics are used to determine the suitability of the compression techniques for different applications. The following sections discuss each of these attributes in more detail, with great focus on the experimental settings of the succeeding chapters.

3.2 Compression Ratio

The effectiveness of a compression scheme is indicated by its compression ratio, which is determined by dividing the amount of data before the compression by the amount of data after the compression. Through the removal of redundancies and sometimes at the expense of fidelity, a compression system reduces the entropy of the video data, thus reducing the bitrates required to store or transmit a bitstream.

The compression ratio can be found from a simple formula which is the size of the original data divided by the size of the compressed image as shown in equation (3.1) below. This ratio shows the capability of different coding algorithms to compress images.

$$Compression\ Ratio = \frac{Size\ of\ Orginal\ image}{Size\ of\ Compressed\ image} \tag{3.1}$$

The compression ratio can be used for indicating the picture quality, since most of the compression techniques operate over a range of compression rate and decompression quality. Generally, the greater the compression ratio, the less the quality of the output images. The trade-off between compression ratio and the quality is an important factor to consider when compressing images.

The amount of data is measured in bits, which is the number of binary symbols required to represent the data. The following bitrates are commonly used to represent video data:

- Bits per frame (bpf)
- Bits per pixel (bpp)
- Bits per second (bps)

To have some idea of the compression ratio required in some common application, table 3.1 illustrates raw bitrates of some common video formats and table 3.2 shows typical target bitrate required by current communications and storage systems.

Format	Size	Colour Format	Frame Rate	Bitrates		
				Per Frame	Per Pixel	Per Second
HDTV	1280×720	4:2:2	60 fps	18.432 Mbpf	20 bpp	1.1 Gbps
PAL	720×576	4:2:2	25 fps	6.6 Mbpf	16 bpp	166 Mbps
CIF	352×288	4:2:0	30 fps	1.2 Mbpf	12 bpp	36.5 Mbps
QCIF	176×144	4:2:0	30 fps	0.304 Mbpf	12 bpp	9.1 Mbps

Table 3.1- bitrates of some common video formats [37]

If a HDTV is taken as an example, in order to transmit 600 Mbps raw-video content through a 20 Mbps channel, the compression system needs a compression ratio of 30: 1. In a video-phone application, a typical video requires a QCIF format at 10 frames per second (fps), which results in a raw bitrate of 3 Mbps; at a channel capacity of 24 kbps, the encoder needs to be compressing at a rate of 125: 1.

Application	Bitrate
POTS Videophone	10 -25 Kbps
ISDN Video Conferencing	384 Kbps
Video CD	1.5 Mbps
Video DVD	2-10 Mbps
WLAN Video	0.1 – 10 Mbps
HDTV	20 Mbps

Table 3.2 - Bitrate required by current communications and storage systems [37]

3.3 Image Quality

Image quality is one of the significant measures for image and video compression systems. Normally, the compression and decompression process cause the degradation of the reconstructed image. So the image fidelity can be used to assess the degree of degradation. The image quality can be grouped into two quality measures: subjective image quality and objective image quality. Subjective image quality is determined by statistically processing the fidelity rating given by a group of human viewers. Objective image quality is defined by a computational process that does not require human intervention. For the subjective quality assessment, the quality is rated using a discrete or a continuous scale ranging from bad quality to excellent quality. Subjective quality assessment is very tedious, expensive and cannot be conducted in real time. Furthermore, it requires many considerations, standard viewing conditions; criteria for observer and scene selection; assessment procedures; and analysis methods. Many observers are needed and the assessments are lengthy, the procedure is therefore very costly. Moreover, it is very difficult to embed it into a practical video processing system because it cannot be implemented automatically. In contrast, objective assessment can compute the image quality automatically and in a relatively short period of time. This is very important for real world applications. For this reason the objective quality assessment measures are used more extensively by researchers and throughout this thesis.

A number of models for objective perceptual video quality and assessments have been introduced over the years. However, the objective methods for image quality estimation can be divided into three types. Full Reference (FR) models where the difference between the original and the distorted sequence is computed. Reduced Reference (RR) models which compute statistics on the distorted sequence and compare them with corresponding stored statistics from the original sequence. Non Reference (NR) models do not use the original sequence at all.

The simplest measures of quality are the Mean-Square Error (MSE) and the Peak Signal to Noise Ratio (PSNR). The MSE between two images is given by

$$MSE = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} (x_{ij} - \hat{x}_{ij})^2 \qquad (3.2)$$

where the images size is $M \times N$, x_{ij} is the original image and $\hat{x}_{ij}$ is the reconstructed image.

One problem with *MSE* is that it depends strongly on the image intensity scaling. In contrast, *PSNR* avoids this problem by scaling the *MSE* according to the image. It is determined as follows:

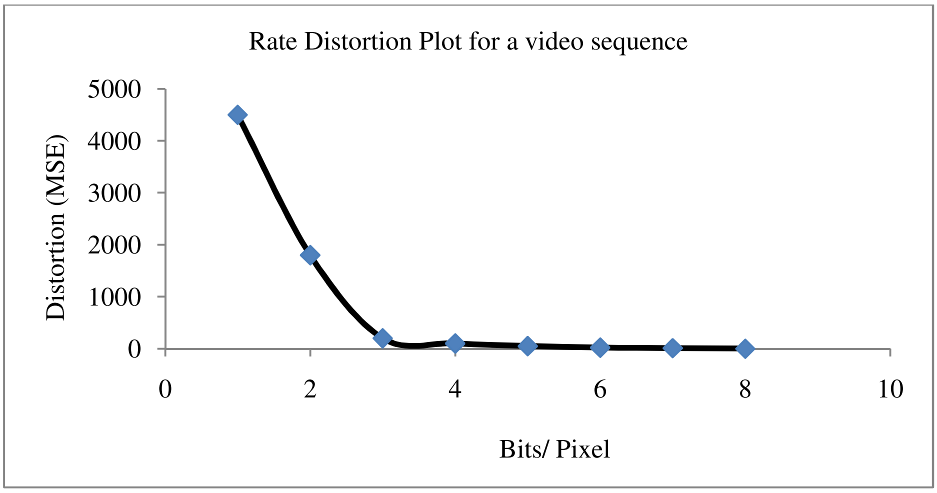
$$PSNR = 10 \log_{10}(\frac{S^2}{MSE}) \qquad (3.3)$$

where, *S* is the maximum intensity value. *PSNR* is measured in decibels (dB). This measure (PSNR) is also not ideal, but it is commonly used. Its main failure is that the signal strength of the image is estimated as $(S)^2$ (value squared), rather than the actual signal strength of the image. However, the difference between PSNR for different compression methods or parameter settings is still a valid comparison, despite this drawback.

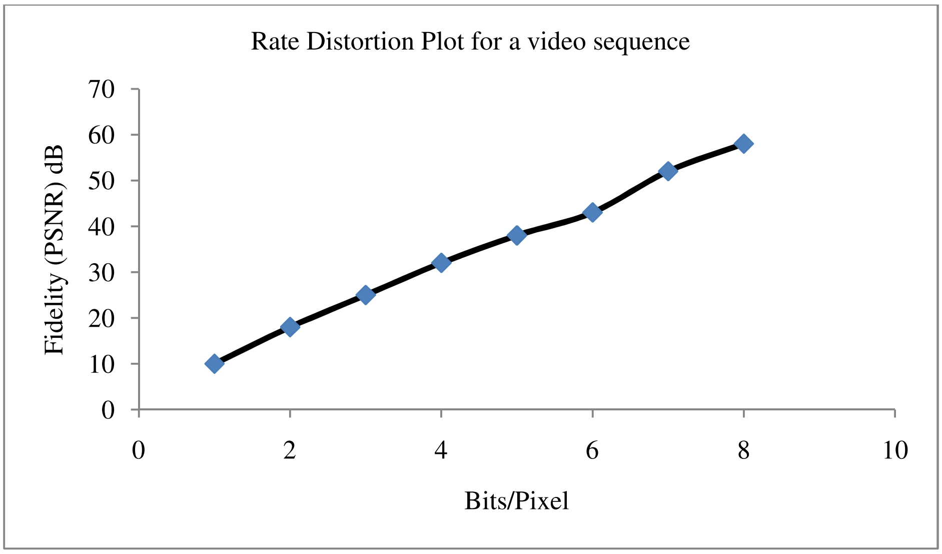
It should be noted that quantitative measures like *MSE* and *PSNR* provide a tangible measure of the amount of distortion introduced by the compression system; they do not take into consideration how the viewers perceive the distorted image. More subjective measures like the mean opinion scores are one of the recent attempts to incorporate subjectivity into the distortion measurements [38].

3.4 Rate-Distortion Theory

Shannon's theoretical analysis of the relationship between fidelity and coding rate [39] has played a significant role in the progress of research in video compression techniques. When the distortion is plotted against the bitrate as illustrated in figure 3.1, the PSNR generally increases as bitrate increases whilst MSE generally decreases as bitrate increases.



(a)



(b)

Figure 3.1 - (a) MSE-bitrate relationship -(b) PSNR-bitrate relationship [37]

A direct use of R-D plot is to estimate the performance of a coding scheme. If the simple algorithm of bit truncation used in figure 3.1 is taken as an example, in order to achieve a PSNR of 30 dB, approximately 4 bits per pixel is needed. Another use of R-D plot is for comparing performances of difference coding schemes. In figure 3.2, two schemes are compared. The R-D curve of (b) lies entirely above that of (a), this indicates that (b) is a better coding scheme than (a). For example at 2 bits/pixel (bpp), scheme b can achieve an improvement of 12 dB over scheme a.

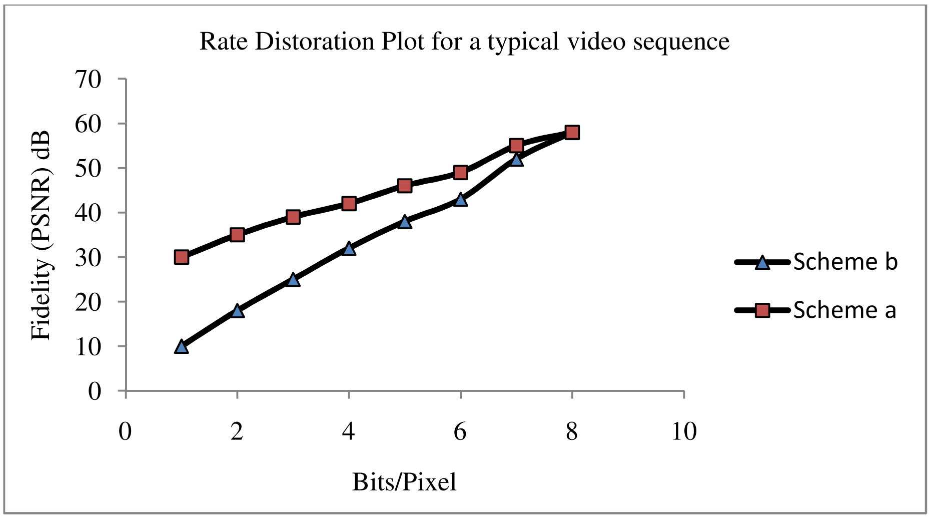


Figure 3.2 - Comparison between two coding schemes

When the performance of low-complexity algorithms is evaluated, typically the rate-distortion performance of the proposed algorithm is plotted, against the rate-distortion performance of the reference encoder. The rate-distortion performances of the two are compared, along with the actual computational complexity savings achieved by the low complexity algorithm.

3.5 Compression Speed

Compression and decompression times are defined as the amount of time required for compressing and decompressing a picture or one image frame. These values depend on the following considerations:

1. The complexity of the compression/decompression algorithm, where a complex compression technique can produce better quality images, but it could be time consuming which is not suitable for some real-time applications.

2. The implementation efficiency of the software or the hardware of the algorithm.

3. The speed of the utilised processor or auxiliary hardware.

Generally, fast compression/decompression techniques are required for real-time applications, whilst for other application speed can be traded for better quality. Fast encoding time increases the speed with which resulting compressed image can be created. Fast decoding times increase the speed with which the user can display and interact with the reconstructed images. Speed of compression is more important if the data is to be transmitted rather than stored, specifically in real-time applications. The decompression speed is important for storage and retrieval and is vital for reception of transmitted data, examples of this include DVD players where the speed of the decompression process is far more important than the speed of the compression speed that normally takes place during the manufacturing.

3.6 Experimental Setup

The experimental setup used in the succeeding chapters for investigating and comparing algorithms can be described by a generic diagram, as shown in figure 3.3

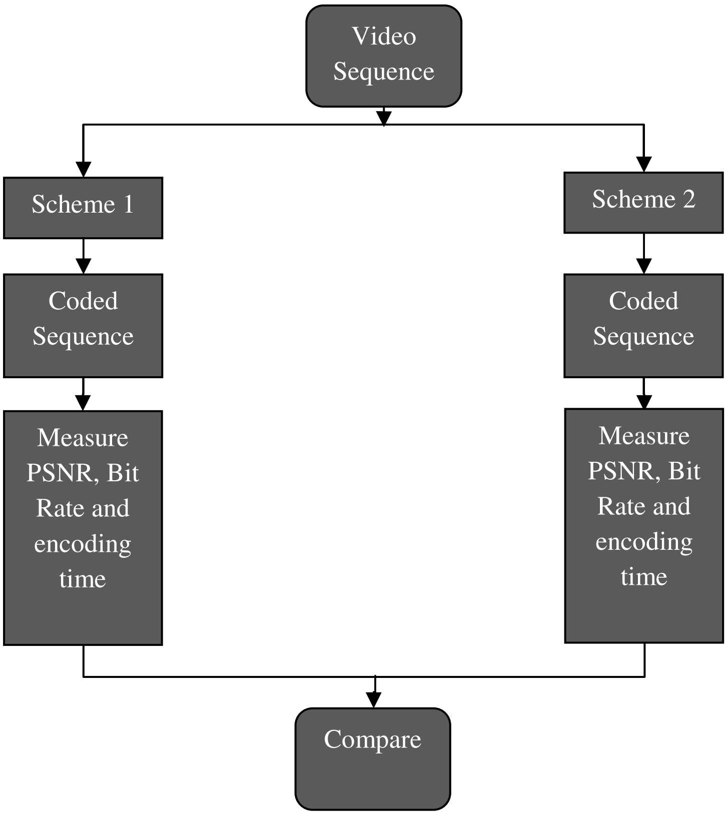

Video Sequence
Scheme 1
Scheme 2
Coded Sequence
Coded Sequence
Measure PSNR, Bit Rate and encoding time
Measure PSNR, Bit Rate and encoding time
Compare

Figure 3.3 - The general experimental setup

3.6.1 Test Environment

An Intel Core(TM) i7 CPU 920 @ 2.47 GHz with 8.0 GB RAM running Windows 7 was used. The Intel VTune performance analyzer [40] was used to measure the number of machine cycles differences, reflecting the total encoding time.

3.6.2 Reference software

In this work, the H.264/AVC reference software codec JM17.2 [31] (referred to as the JM codec), is used as the reference video codec. The JM codec is commonly used to test new algorithms in the video coding community. The use of this reference software enables realistic comparison of the performance of different algorithms developed by different researchers. The source code (in the C programming language) for the JM codec can be downloaded from [31] The earlier and the later versions of the JM codec and the revised manual [41] for the H.264 reference software can also be found from [31].

The JM encoder reads input parameters from a configuration file. A wide range of encoding parameters can be changed using the configuration file. These include but are not limited to:

- Input video sequence (concatenated YCrCb 4:2:0 format)
- Quantisation parameters for I, P and B slices
- Available MB partition modes
- I, P and B picture sequence
- Number of reference frames
- Rate-distortion optimisation - ON/OFF

The JM codec also provides useful encoding statistics such as bitrate of the encoded bitstream, video quality in PSNR of luminance and chrominance components of the coded video and encoding time.

For the Scalable and the Multiview extensions similar software has been used [42, 43] both of these are based on the JM software but written in C++.

3.6.3 Development Environment

The source code is compiled and built using Microsoft Visual C++ Professional Version 9.0. This development environment is used to modify the source code to incorporate the algorithms to be tested.

3.6.2 Test Video Sequences

Test sequences are chosen from the JVT recommended test video materials widely used by researchers and scientists with the goal of covering a range of content detail, object motion and different types of background as well as camera movement. All sequences are in YUV 4:2:0 colour formats, in which the two chroma components are downsampled by a factor of two in each spatial direction.

H.264/AVC Test Video Sequences

Sequence	Size	Characteristic
Akiyo	CIF QCIF	A sequence with slow motion and fixed background.
Foreman	CIF QCIF	A sequence with medium changes in motion and contains dominant luminance changes.
Tempete	CIF QCIF	A sequence of with highly detailed spatial content, fast random motion and camera zoom.
Silent	CIF QCIF	A sequence of low spatial details and medium changes in the motion of the arms and head of the person in the sequence.
Stefan	CIF QCIF	Contains panning motion and has distinct fast changes in motion.
Mobile	CIF QCIF	Contains slow panning, zooming, a complex combination of horizontal and vertical motion and high spatial colour detail.

Table 3.3 - H.264/AVC test video sequences

H.264/SVC Test Video Sequences

The test sequences that are used for simulations for the H.264/SVC are summarised in tables 3.4. The tables specify the maximum spatial and temporal resolution of the sequences. Sequences with a lower temporal resolution are obtained by frame skipping, and sequences with a lower spatial resolution are obtained by downsampling as specified in the JSVM [42].

The test sequences are classified into a high-delay and a low-delay test set. The high-delay test set contains sequences, which have been widely used for testing purposes during the SVC development. The sequences in this set contain different amounts of detail and motion. Since low-delay is mainly required for interactive video telephone or videoconferencing applications, the low-delay test set consists of a variety of video conferencing sequences.

Sequence	Size	Frame Rate	Characteristic
City	CIF QCIF	7.5,15,30	high-delay
Foreman	CIF QCIF	7.5,15,30	high-delay
Mobile	CIF QCIF	7.5,15,30	high-delay
Harbour	CIF QCIF	7.5,15,30	high-delay
Silent	CIF QCIF	7.5,15,30	low-delay
News	CIF QCIF	7.5,15,30	low-delay
Hall	CIF QCIF	7.5,15,30	low-delay

Table 3.4 - H.264/SVC test video sequences

H.264/MVC Test Video Sequences

Source	Sequences	Image Property	Camera Arrangement
MERL	Ballroom, Exit	640x480, 25fps	8 cameras with 20cm spacing; 1D/parallel
HHI	Uli	1024x768, 25fps	8 cameras with 20cm spacing; 1D/parallel convergent
KDDI	Race1	640x480, 30fps	8 cameras with 20cm spacing; 1D/parallel
KDDI	Flamenco2	640x480, 30fps	5 cameras with 20cm spacing; 2D/parallel (Cross)
Microsoft	Breakdancers	1024x768, 15fps	8 cameras with 20cm spacing; 1D/arc
Nagoya university / Tanimoto Lab	Rena	640x480, 30fps	100 cameras with 5cm spacing; 1D/parallel
Nagoya university / Tanimoto Lab	Akko&Kayo	640x480, 30fps	100 cameras with 5cm horizontal and 20 cm vertical spacing; 2D array

Table 3.5 - H.264/MVC test video sequences

Table 3.5 describes the properties of the various test data sets [44]. These data sets vary in the number of cameras/views, the arrangement of the cameras, distance between cameras, as well as properties of the images in terms of image size and frame rate. All sequences are provided in YUV 4:2:0 planar format, except for the Microsoft Research data that is available in BMP which was then converted to YUV.

3.6.3 Computational complexity measurement

In this work three different methods have been chosen to calculate complexity depending on the different experiments.

Encoding time

Encoding time is a direct measurement of the algorithm complexity in software-only encoders. In order to estimate the complexity of different types of encoder, the time spent on encoding video sequence (in milliseconds) is recorded and utilised to compare the complexities of different algorithms. This is the most commonly used method by researchers in the field of video compression.

Number of processed blocks

Encoding time depends on the processing speed of the workstation used for simulation. The time cost for a single function during encoding a single frame is very small, therefore, it is difficult to measure accurately the complexity of each function using encoding time. In Chapter 5, Chapter 6 and Chapter 7, the number of MBs or blocks actually coded is chosen as an assessment of computational complexity. For example when early termination algorithm is proposed the ratio of the terminated MBs is used as an indication to the percentage saving of time.

Machine Cycles

The third way used to calculate the computational complexity is the Intel VTune performance analyzer [40]. It measures the number of machine cycles of each of the individual functions whilst encoding a video sequence. This provides accurate information about processor utilisation. The extra complexity of each proposed algorithm is calculated in terms of basic operations used in the computer, including addition, multiplication, shift and comparison.

3.6.4 Performance Evaluation

In [45] a technique to express RD plots where PSNR and bitrate differences between two simulation conditions can be read was proposed. The method for calculating the average difference between two such curves was presented using the following steps.

1. Fit a curve through 4 data points (PSNR/bitrate are assumed to be obtained for QP = 16,20,24,28)

2. Based on this, find an expression for the integral of the curve

3. The average difference is the difference between the integrals divided by the integration interval

In [46], Bjontegaard realised that interpolation methods normally used lead to domination of the high bitrates to the difference between the curves. Therefore, integration based on logarithmic scale of *bitrate* was suggested. With a logarithmic bitrate scale, the *PSNR* can be calculated using third degree polynomial as shown in equation 3.4

$$PSNR = a + b \times bitrate + c \times bitrate^2 + d \times bitrate^3 \qquad (3.4)$$

In the same way the interpolation can be used to find the *bitrate* as a function of *PSNR*, shown in equation 3.5

$$Bit = a + b \times PSNR + c \times PSNR^2 + d \times PSNR^3 \qquad (3.5)$$

Using this method the following can be found:

- Average PSNR difference in dB over the whole range of bitrates, this is known as the Bjontegaard Delta PSNR (BDPSNR)

- Average bitrate difference in % over the whole range of PSNR, this known as Bjontegaard Delta BitRate (BDBR) percentage differences.

Along with the encoding time those two measurements are widely used amongst researchers to compare video coding algorithms. Thus, this approach has been used throughout the thesis.

3.7 Summary

This chapter presents the experimental methods used in this research project. The algorithms developed during the project are tested using software simulation. The H.264 reference software code is used as the reference codec and the algorithms are implemented by modifying the source code of the reference encoder. Standard test video sequences are used to evaluate the performance of the algorithms. The performance of the algorithms is evaluated by measuring the computational complexity, bitrate and objective video quality.

CHAPTER 4
FAST MODE DECISION FOR THE H264/AVC

4.1 Introduction

Chapter 2 illustrated that the Rate Distortion Optimisation (RDO) process is coupled with the selection of the best coding mode for each MB (16×16 pixel area). The best mode is the block size used to perform ME with the minimal Rate Distortion (RD). This is done in order to achieve high performance compression and image quality but at the expense of increased encoding complexity. Consequently several fast Mode Decision (MD) and ME techniques have been developed to reduce the computational cost. These approaches successfully reduce the computational time by reducing the image quality and/or increasing the bitrate. In this chapter a novel fast MD and ME technique is proposed. The algorithm utilises pre-processing frequency domain ME in order to accurately predict the best mode and the search range. In this work the correlation between the 16×16 MB and its prediction is measured, and based on the result the best mode is selected, or the mode selection process is limited to a subset of modes. Moreover using the correlation result an appropriate search range for the ME stage is selected.

Experimental results show that the proposed algorithm significantly reduces the ME time by up to 97% whilst maintaining similar RD performance when compared to the JM software [47, 48].

The rest of the chapter is organised as follows. Section 4.2 presents observations on recent work in the area, while Section 4.3 describes the proposed mode decision algorithm. Section 4.4 contains a comprehensive list of experiments and a discussion of the results. Section 4.5 summarises the chapter.

4.2 Observations on Previous Work

4.2.1 Overview

There are three main categories of fast mode decision schemes; fast skip and direct mode decision, fast inter-mode decision and fast intra mode decision [49]. The work in this chapter is focused on increasing the speed of the skip and inter-mode decision, therefore, in the following sub-sections the approaches to fast mode decision in those categories are discussed and compared.

4.2.2 Background

Before describing techniques for fast mode decision the concept of predicted MV needs to be outlined. Due to the high correlation of the MV field in adjacent MBs, the MV(s) of the current MB can be predicted from vectors of previously encoded adjacent MBs. This leads to only encoding the difference between the optimal MV after the ME, and the predicted MV [50]. By encoding the Motion Vector Difference (MVD) a significant number of bits can be saved. Figure 4.1 shows different neighbouring MBs that are used to predict the MVs of the current MB (D in the figure).

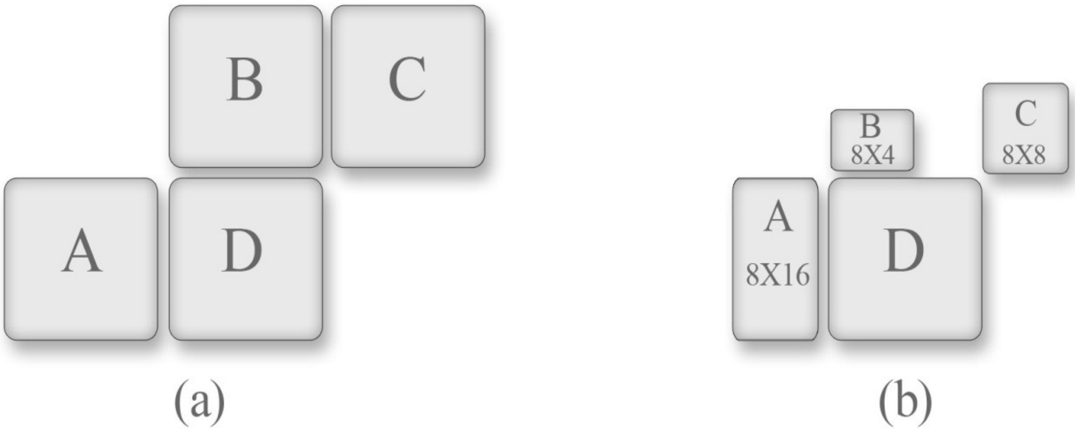


Figure 4.1- (a) Current block and neighbour blocks have the same partition (b) Current block and neighbour blocks have different partition

Predicted and MVD of the current block (MV_p and MV_d respectively) are calculated using equation (4.1):

$$
\begin{aligned}
MV_P &= \frac{MV_A + MV_B + MV_C}{3} \\
MV_d &= MV_R - MV_P
\end{aligned}
\qquad (4.1)
$$

where MV_R is the optimal motion vector for the current block after motion estimation.

In areas that do not have any motion the H.264 allows the motion to be predicted using the skip mode [10]. The MV of skip mode is generated identically to the prediction MV for the 16×16 motion compensated MB mode. Using this mode no motion or texture information are sent to the decoder, thus coding efficiency can be improved especially for slow video sequences.

The direct mode was firstly introduced in [51], it exploits the temporal correlation by bi-directional prediction from both forward and backward reference pictures. This mode does not require any bits for coding the motion vectors but texture information needs to be sent to the decoder. The direct mode uses two motion vectors; forward motion vector MV_0 and backward motion vector $MV_{1,}$ both are generated from the motion vector MV_C of the co-located MB of the next reference picture $RF_{1.}$ The predicted best match is calculated by the linear combination of the two blocks which have motion vectors MV_0 and MV_1 and are located on the two reference frames RF_0 and RF_1 respectively. Figure 4.2 along with equation 4.2 illustrate the generation of motion vectors in the direct mode.

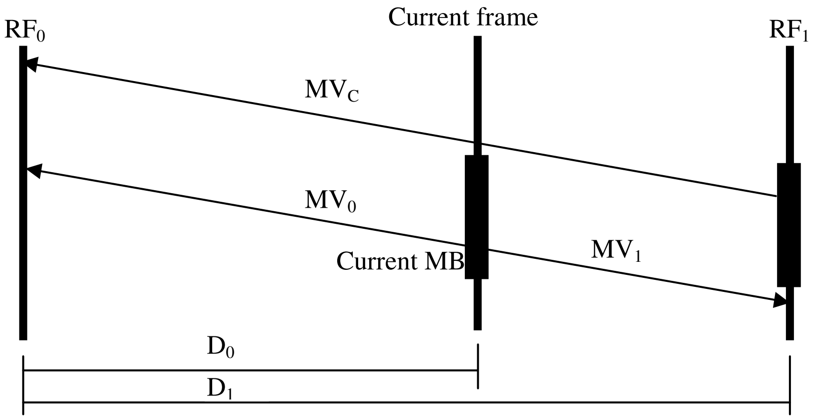


Figure 4.2 - MV_0 and MV_1 of a direct mode

$$MV_0 = \frac{D_0}{D_1} MV_C \qquad MV_1 = \frac{D_0 - D_1}{D_1} MV_C \tag{4.2}$$

where D_1 is the temporal distance between the reference frames $RF_{0,}$ $RF_{1,}$ and D_0 is the distance between the current frame and the reference frame $RF_{0.}$

4.2.3 Fast skip/direct mode decision techniques

The authors in [52] proposed a skip mode detection technique, it utilises the value of SAD between the current MB and the co-located MB to determine skipped MBs. If SAD for a given QP is smaller than an adaptive threshold, then the current MB is skipped. This skip mode detection despite using adaptive thresholds does not consider the predicted motion vectors, thus the proposed algorithm leads to very little improvement if this is taken into consideration.

A fast skip mode decision technique based on spatio-temporal neighbourhood information was presented in [53]. The main idea is based on the observation that if for a MB has the skip mode as its best mode; it is very likely that the co-located MBs in any temporal direction will have similar mode. Moreover, if the two MBs on the top and left of the one to be encoded in the current frame and the MBs on the right and bottom of the co-located MB in the previous frame are all in skip mode, then this mode pattern can be a good indication that the current MB can be skipped. To strengthen the accuracy of this skip mode prediction, an extra condition was added that the SAD between the current MB and its co-located one should be less than the average SAD among the skipped MBs in the reference picture and their co-located predictors. The proposed scheme can predict skip mode without any ME thus significantly saving computational complexity for very similar RD performance to the standard.

Two observations regarding this scheme are that; the performance is totally dependent on the video content and that this algorithm can be improved since no predicted motion vector information was considered.

The skip mode decision is affected by several factors such as the choice of QP and the Lagrangian cost evaluation which makes the schemes proposed by [52] and [53] dependent on video content and coding conditions.

In [54] a novel hierarchical skip mode detection technique was proposed. Considering that skipped MBs tend to occur in clusters, spatial and temporal skip mode information is used in the first layer. If the co-located MB in the RF or at least one of the upper or the left MB of the current MB in the current frame is in skip mode, the current MB passes through the first layer. Otherwise the current MB cannot be predicted as skip mode directly. In the second layer, if the SAD between the current MB and its co-located MB in the RF is smaller than an adaptive threshold the current MB can be considered as a potential skip MB. Otherwise the current MB cannot be predicted as skip mode. A fast scheme based on [55] for identifying transformed-quantised coefficients which have become zero is used in the third layer. Finally, the current MB is in skip mode if it passes through all the three layers. Despite the authors including the QP and the H.264/AVC integer transform cues in the mode decision process as opposed to [52] and [53], the predicted motion vector information is still not considered in their scheme.

Another skip mode decision scheme was proposed in [56] in which the skip mode RD cost of the current MB is calculated using a model based on local sequence statistics and for a given Lagrange multiplier. The early skip mode decision is made by comparing the latter RD cost to the predicted one. The achievable computational savings for typical video sequences are in the range of 19%-67% without significant loss of RD performance.

A fast skip mode decision technique has resulted in contributions to the standard, was proposed in [57]. According to this work, a MB can be predicted as having skip mode in the baseline profile, when the following set of four conditions is satisfied:

1. The best motion compensation block size for this MB is 16×16 (MODE_16×16)

2. The best reference slice is the previous slice

3. The best motion vector is the predicted motion vector (regardless of this being a zero motion vector or a non-zero one)

4. The transform coefficients of the 16×16 block size are all quantised to zero.

This set of four conditions is not sufficient due to the assumption that the mode with the lowest RD cost J is the inter-mode_16×16 (condition 1), which may be valid or not. If it is true, then it is safe to claim that the MB can be skipped since $J_{SKIP} < J_{MODE\_16\times16}$ for the same MVs (satisfy condition 3 above). If condition 1 is not true, the algorithm will incorrectly predict MBs as skipped and the RD performance will suffer. However, this algorithm only needs to perform ME for the 16×16 mode and thus save computation for the remaining mode types given that the above conditions are satisfied. Experimental results showed that this method reduces computation complexity by 15% on average depending on different video sequence without any noticeable RD performance loss.

Similar ideas for predicting direct and skip modes for B slices were also presented in [58]. A MB is predicted in skip mode when the following set of two conditions is satisfied:

1. The reference slices and the MVs are the same as the ones decided under the direct mode.

2. The transform coefficients of the 8×8 sub-blocks of this MB are all quantised to zero.

In summary skip/direct modes are especially useful for low bitrate coding and their early detection can lower the encoder complexity significantly with only small losses in quality. Most skip/direct mode decision techniques exploit temporal and spatial neighbourhood information in combination with adaptive thresholds. The more advanced techniques in this category also incorporate the QP and the inferred motion vector(s) of the standard to minimise further the impact on the RD performance [49].

4.2.4 Fast Inter-Mode Decision

A fast inter-mode decision method which only depends on the MAD between the current and previous frames was proposed in [59]. The main idea is to use large block types for smooth areas and small blocks for areas containing complex motions. If the MAD between the current and corresponding MB(s) in the reference frame(s) is smaller than the weighted Mean Absolute Frame Difference (MAFD) between the current and previous frames, only large mode types from the set {16×16, 16×8, and 8×16} are chosen. Otherwise all mode types are examined. The disadvantage of this algorithm is that the weights are derived for a fixed QP, thus the scheme is not very practical in rate controlled applications where the QP changes per MB. The proposed algorithm can obtain up to 48% computational savings with similar RD performance as the standard.

The SAD has been used in a similar manner to predict the best mode in [60]; the scheme achieves 40.73% computational cost reduction with 0.04dB PSNR degradation and 0.92% bitrate increase for a set of test sequences.

In another set of algorithms, cost metrics based on the SATD have been proposed in [61] and [62]. These SATD based algorithms are similar to the low complexity mode decision schemes in the standard (see appendix A) but used as an indicator of optimal mode in a high complexity scheme. The three most probable modes with lowest costs in low complexity mode decision are chosen for high complexity mode decision. The drawback of this idea is that mode candidates are known only after all ME has been performed in the low complexity mode, thus only part of the mode decision process can be saved. About 40% time savings in the high complexity mode of H264/AVC can be observed for an average PSNR loss of 0.07dB as compared to the standard.

In [63] the MV information has been used to predict the inter-modes, however a maximum bitrate increase of about 15% occurs at the same quality as the standard, thus the RD performance is degraded rather significantly.

A fast inter-mode decision algorithm is proposed in [64] which exploits the correlation of the different modes' *J* costs. The basic idea of this algorithm is that if the cost of larger block-size modes is higher than the cost of the current block-size mode, then the best mode of current MB cannot be of a larger block-size. Meanwhile, if the cost of a smaller block-size mode is higher than that of current block-size mode, then best mode for the current MB cannot be of a smaller block-size.

A fast inter-mode decision based on fuzzy classification theory [65] was proposed in [66]. The scheme exploits spatio-temporal correlation. There are two categories in temporal correlation, namely complex motion MB and simple motion MB. Different inter-mode candidates are assigned to the two categories, based on early termination techniques for ME.

In [67], the authors proposed a fast mode decision algorithm using a filter bank of Kalman filters. Firstly, the current MB is encoded using a pre-determined set of candidate modes. If the minimal RD cost of the current MB is less than the average RD cost of the modes of previously encoded MBs in the current slice, a small set of candidate mode are assigned to the current MB. Otherwise, a large set of candidate mode types is assigned. The main difference in this algorithm in comparison to other algorithms is instead of using temporal information spatial information is used. In this work, a time reduction of about 30% was claimed with only small degradations in video quality compared to the standard. Since this algorithm was designed especially for HD video encoding, it cannot be used for low bitrate and/or low resolution video sequences because the collection time for filter evaluation is prohibitive, thus the time savings will be insignificant for similar RD performance as the standard.

In summary, fast inter-mode decision algorithms can achieve time savings in the range of 30% - 80% for similar RD performance to the standard. Due to the high diversity of these techniques and the different experimental settings used for measuring complexity and RD performance, it is very difficult to claim that one class of techniques performs better than another. However, mode decision techniques using a statistical model to predict modes tend to report slightly higher gains of time savings for similar RD performance to the standard. Furthermore, the idea of joint optimisations for ME and mode shows very good results.

Finally, the relatively new and unexplored area of incorporating low complexity computer vision principles for accurately estimating motion and detecting changes in illumination in the mode decision problem may provide more significant time savings for similar performance with the standard, since ME accuracy and illumination variations seem to affect significantly the mode decision process.

4.3 Proposed Algorithm

Recently there has been a lot of interest in ME techniques operating in the frequency domain. These are commonly based on the principle of cyclic correlation and offer well-documented advantages in terms of computational efficiency due to the employment of fast algorithms. One of the best-known methods in this class is phase correlation [13] which has become one of the ME methods of choice for a wide range of professional studio and broadcasting applications [68]. In addition to computational efficiency, phase correlation offers key advantages in terms of its strong response to edges and salient picture features, its immunity to illumination changes and moving shadows and its ability to measure large displacements. Several attempts [69, 70] have been proposed to adapt the phase correlation to the H.264 standard. In [69] the authors proposed an adaptive block size phase correlation ME which has been compared to the Full Search Block Matching (FSBM) algorithm [31], however this method significantly increases the bitrate when compared to the standard. Furthermore block sizes up to 32×32 were used to estimate the motion which increases the computational complexity. In [70] the authors used the phase correlation to predict the ME block size by generating a binary matrix, and then selecting the mode from the binary matrix. Although the authors claimed 50% reduction in the ME time, the algorithm showed significant RD performance degradation for slow video sequences.

In video compression, knowledge of motion helps to exploit similarity between nearby and adjacent frames in the sequence, and remove the temporal redundancy between neighbouring frames in addition to the spatial and spectral redundancies [71]. The phase correlation method measures the movement between the two fields directly from their phases. The basic principles are described below.

Assuming a translational shift between the two frames:

$$s_t(x, y) = s_{t+1}(x + \Delta x, y + \Delta y) \tag{4.3}$$

where s_t is the current frame and s_{t+1} is the following frame.

Their 2-D Fourier transforms are:

$$S_t(f_1, f_2) = S_{t+1}(f_1, f_2)\exp[2j\pi(f_1\Delta x + f_2\Delta y)] \tag{4.4}$$

Therefore the shift in the spatial-domain is reflected as a phase change in the spectral domain. The cross-correlation between the two frames is:

$$C_{t,t+1}(f_1, f_2) = S_{t+1}(f_1, f_2) \cdot S_t(f_1, f_2) \tag{4.5}$$

The normalised cross-power spectrum is:

$$R_{t,t+1}(f_1, f_2) = \frac{S_{t+1}(f_1, f_2) \cdot S_t^*(f_1, f_2)}{\left|S_{t+1}(f_1, f_2) \cdot S_t^*(f_1, f_2)\right|} \tag{4.6}$$

From equations (4.4) and (4.6):

$$R_{t,t+1}(f_1, f_2) = \exp[-2j\pi(f_1\Delta x + f_2\Delta y)] \tag{4.7}$$

The 2-D inverse transform is given by:

$$c_{t,t+1}(x_1, y_1) = \delta(x_1 - \Delta x, y_1 - \Delta y) \tag{4.8}$$

The displacement can be found by using the location of the pulse in equation (4.8). The maximum correlation, achieved when the two images are identical, has a value of 1 at (0, 0).

Observation on the phase correlation results for different images extracted from different video sequences (details of those sequences are given in chapter, table 3.3) revealed that if the correlation between the MB and its prediction is greater than or equal to 0.8, 92% of the time the MB contains objects that have minimum size of 16×8 or 8×16 and the MV has a maximum value of 8 in any direction. On the other hand when the correlation is less than 0.8 this indicates that contents of the MB are either large objects with large movements or number of small objects with various movements. This threshold has been chosen to control the RD performance when compared to the standard, furthermore, limiting the search range to 8 reflects the maximum movement that a 16×8 or 8×6 object can have in a 16×16 area whilst maintaining high correlation (i.e. the movement is inside the 16×16 macroblock).

Using the above observation the following algorithm [47, 48] has been developed: if the correlation value is equal to 1, then the skip mode is chosen as the best mode. Otherwise if the correlation value is greater than or equal to 0.8, the mode selection process is limited to modes {0, 1, 2 and 3}, additionally the search range is limited to 8. Finally if the correlation value is less than 0.8, all the modes are enabled and ME is preformed using the defined search range. The proposed algorithm is shown in flowchart form in figure 4.3. The phase correlation implementation is adapted from openCV.

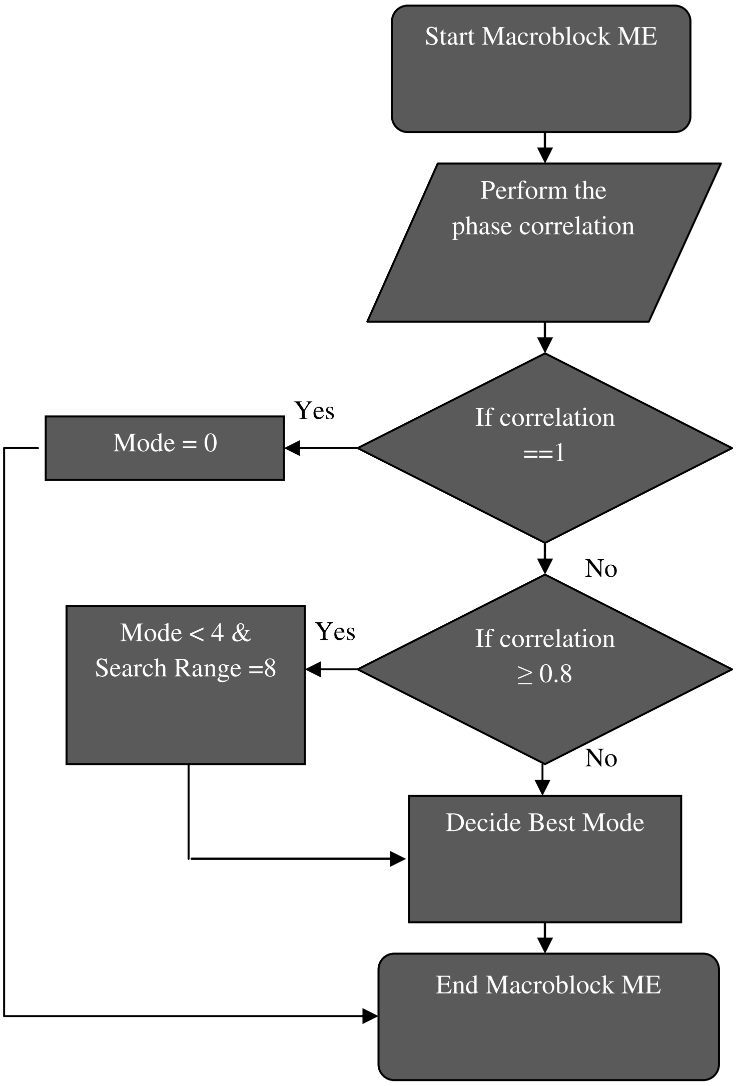


Figure 4.3- The proposed algorithm

4.4 Experimental Results

To assess the proposed algorithm, a comprehensive set of experiments for six kinds of video sequences with different motion characteristics was performed. Details of those sequences and the experimental setup condition can be found in chapter 3.

The encoder configuration settings have the number of reference frames as 1 and the chosen search range was 32 pixels for the full motion estimations.

Table 4.1 shows the percentage cycle savings, the percentage search point savings, the Bjontegaard Delta BitRate (BDBR) percentage differences and the Bjontegaard Delta PSNR (BDPSNR) differences (in dB) between the JM software [31] and the proposed new algorithm [47, 48].

Sequence	Size	BDPSNR (db)	BDBR (%)	Cycles Saving (%)	JM ME Time (s)	Proposed Algorithm ME Time (s)	ME Time Saving (%)
Akiyo	QCIF	-0.08	+1.3	66.98	120.3	3.4	97.02
	CIF	-0.04	+0.96	57.36	286.6	39.45	86.49
Foreman	QCIF	-0.05	+1.22	36.43	272.3	146	45.17
	CIF	-0.04	+1.14	35.74	733.9	408.94	44.68
Tempete	QCIF	-0.01	+0.61	39.75	401.61	222.4	44.31
	CIF	-0.05	+0.76	40.07	985.24	526.53	46.8
Silent	QCIF	-0.01	+0.36	60.53	180..48	34.48	80.9
	CIF	-0.03	+0.77	52.69	508.2	126.9	75.02
Stefan	QCIF	-0.03	+0.3	30.54	384.63	245.14	36.53
	CIF	-0.04	+0.5	29.39	914.75	586	35.81
Mobile	QCIF	-0.02	+0.2	26.06	519.63	341.5	34.88
	CIF	-0.05	+0.6	27.4	1152.6	770.9	32.74
Average		-0.04	+0.7	41			55

Table 4.1- Comparison between the proposed algorithm and JM software

The above table shows that the bitrate percentage differences (BDBR) are in the range of [0.2, 1.3], while the Delta PSNR (BDPSNR) differences are in the range of [-0.08 - 0.01]. The minus signs denote PSNR *degradation* and bitrate *savings* respectively.

This clearly shows that the proposed algorithm had very similar RD performance to H.264/AVC. Furthermore, ME time savings up to 97% and percentage cycle savings up to 67% are observed. It also can be seen that the reduction in the CPU cycles depends on the characteristics of the image sequences. For a slow image sequence with a simple background, the reduction is much more significant than for a fast image sequences or sequences with a more complex background. The reason for this is that in slow video sequences the number of big block sizes increases significantly. A graphical representation of the RD outcome is also provided in figures 4.4 and 4.5 for two sample video sequences.

Moreover, when comparing the results to the results in [70], in addition to the significant time reduction gain, the RD performance is maintained similar to the JM software for the various sequences, whilst in [70] the performances has been degraded rather significantly for some of the sequences.

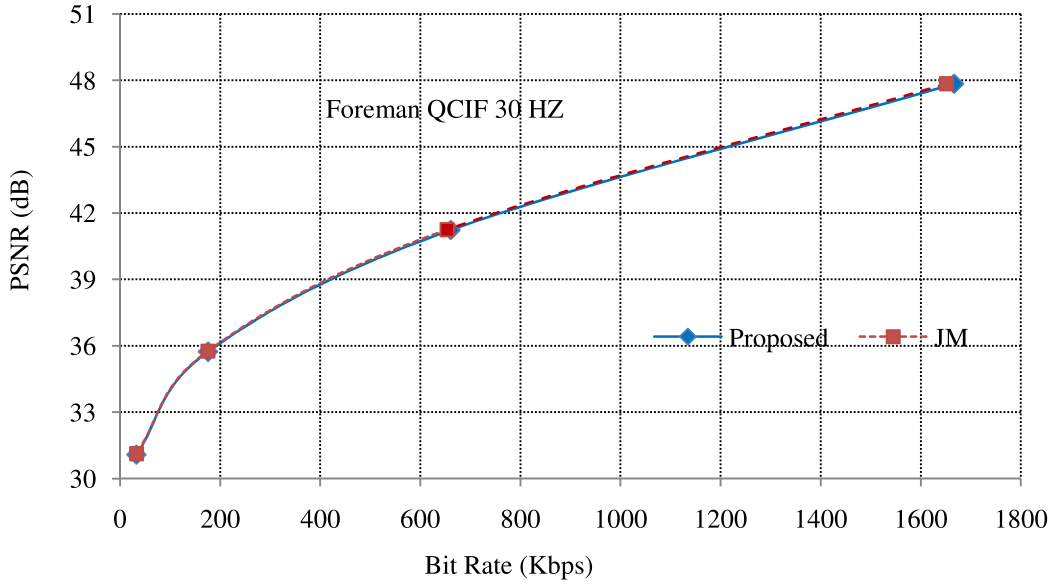


Figure 4.4- RD performance comparison between proposed algorithm and mode the decision algorithm in the JM reference software

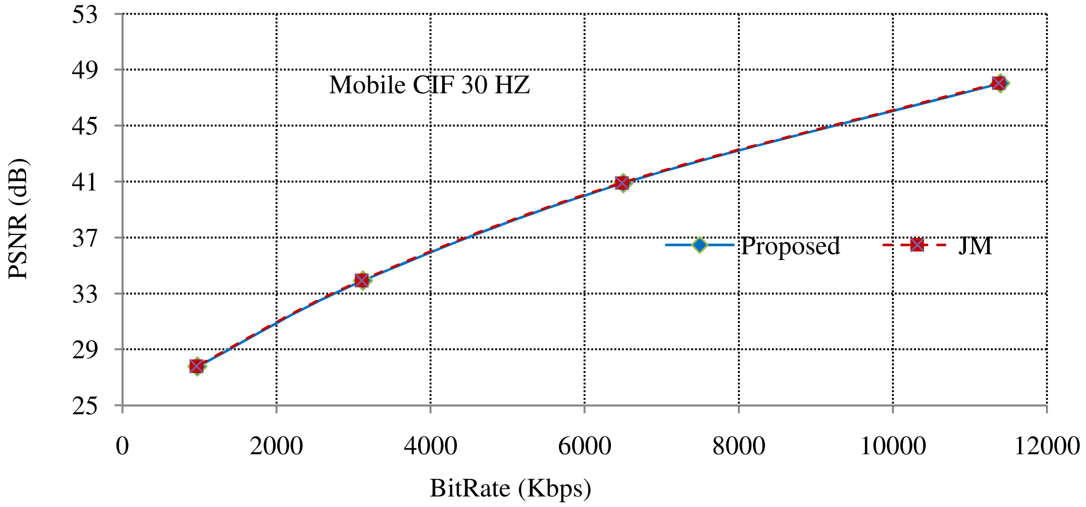


Figure 4.5- RD comparison between proposed algorithm and the mode decision algorithm in the JM reference software

4.5 Summary

The H.264/AVC increases memory bandwidth and spends a significant amount of processing time for the ME process in order to determine the optimal MV. As a means of increasing the coding efficiency, in this chapter, a fast mode decision and ME scheme with RD performance similar to the standard has been proposed. This technique can reduce up to 97% of the ME Time (67% in CPU cycles), resulting in significant time/cycle savings as compared to H.264/AVC. It is very relevant to low complexity video coding systems.

This fast frequency domain ME method was developed as a novel more efficient alternative to the standard ME methods. This method enables the user to encode video sequence within a short period of time. Computational savings are achieved by identifying, prior to the ME and mode decision process, the MBs that are likely to be skipped or to belong to a set of modes and therefore saving further computational processing of these MBs. The early MB mode prediction is made by applying the phase correlation ME before the MB is processed. If the estimated correlation is equal to one the MB is marked as skip, no further processing is carried out for this MB. If the correlation is greater than a threshold, the MB is considered to belong to a large object and only the RD cost of a set a modes is obtained, otherwise the MB is coded as normal.

The advantages of block matching and frequency domain ME techniques have been utilised in the designing of this optimised encoder. The frequency domain ME advantage is shown by the significant saving in the encoding time, while the advantages of block matching ME is shown by maintaining similar bitrate to the standard.

The algorithm main advantage includes a significant time saving which is achieved when compared to the standard. The disadvantage of this algorithm is; complexity savings mainly depend on the activity of the sequence and are therefore unpredictable. The complexity saving is high for low activity sequences because more MBs can be skipped or have partial processing without adversely affecting the rate-distortion performance and vice versa.

In the next chapter the exploitation of another frequency domain based ME algorithm is investigated. Particularly the effect of the Hadamard transform on the different stages of the H.264 recommended ME method is outlined.

CHAPTER 5

Effect of the Hadamard Transform on Motion Estimation of Different Layers in Video Coding

5.1 Introduction

Previously, the motion-compensated prediction result that provides the least distortion was broadly accepted as the prediction signal. However, in recent years, it has been realised that such a selection is not always the most efficient, since the minimal distortion may result in a high bitrate, thereby degrading the overall coding performance [72]. To solve this problem, the Rate-Distortion Optimisation (RDO) concept has been introduced. RDO techniques minimise the distortion under a constraint on the rate. A classical solution to the RDO problem is the Lagrangian optimisation which is used in the H.264/AVC standard. This technique works by converting the RDO problem from a constrained problem to an unconstrained problem as shown in equation 5.1 (also this Lagrangian equation has been shown in chapter 2- equation 2.3)

$$J_{motion} = D_{motion} + \lambda_{motion} R_{motion} \tag{5.1}$$

The Lagrangian cost function is divided into two parts; distortion and rate. The distortion measurement measures the quality of the reconstructed pictures while the rate quantifies the bits needed to code the MB. The Lagrange multiplier is usually calculated in a heuristic way or in an analytical way based on Rate-Distortion (R-D) models [72, 73]

The JM [31] software allows the user to select the ME distortion metric between the Sum of Absolute Differences (SAD), the Sum of the Squared Differences (SSD) and the Sum of Absolute Transformed Differences (SATD), the latter uses the Hadamard transform. This has been employed to improve the RD performance and to facilitate the standard to gain much support in a variety of application areas. By default the JM software selects the SAD as the error metric for full-pixel (first layer) ME and the SATD as the error metric for half and quarter-pixel (second and third layer respectively) ME.

Although SATD is much slower than SAD, it more accurately predicts quality from the standpoint of both objective and subjective metrics.

In this chapter, the implications of the SATD based ME on different layers are discussed. Moreover, a comparison between the SAD and SATD effect on the coefficients' bits and MV bits on different layers is presented and analysed [74].

The chapter is organised as follows. Section 5.2 describes the implications of the SATD based ME on different layers. Section 5.3 presents details and discussion of a comprehensive list of comparative experimental results. Section 5.4 summarises the chapter.

5.2 The Implications of the SATD-Based Motion Estimation on Different Layers

5.2.1 Sub-pixel motion estimation in video compression

Motion-compensated Prediction (MCP) is one of the important elements that fundamentally contributed to the success of the modern video coding standards. It eliminates the temporal redundancy in video sequences and decreases the size of bitstreams significantly. Using MCP, the pixels to be encoded are predicted from the temporally neighbouring ones, and only the prediction errors and the Motion Vectors (MV) are encoded and transmitted.

However, for a number of reasons the positions in between the integer-pixels, called sub-pixels, should be interpolated in order to increase the resolution of MV to enhance the performance of ME, i.e. using sub-pixel ME. Some of the key reasons are as follows:

1. Sampling: converting analogue signal into digital signal through sampling leads to the creation of values at certain points, called integer-pixel positions. However, moving objects between two frames do not always have integer-pixel displacement, therefore, a better match could be obtained by sub-pixel ME at interpolated positions.

2. Lighting condition: slight change in the lighting conditions can lead to finding a better match for the current MB using sub-pixel ME. This is evident for both moving and stationary objects.

3. Noise: noise can generate sub-pixel motion; this is true even for stationary background.

4. Camera shaking: camera shaking can bring motion effect to stationary objects. Thus in those situations, the stationary object may also have better match using sub-pixel ME.

Sub-pixel refinement can greatly improve the performance of ME in terms of both compression ratio and decoded image quality. Experimental results have shown an average of 25% reduction in bitrate and 0.3dB increase of PSNR when sub-pixel ME is enabled.

5.2.2 Interpolation in H.264/AVC

Quarter-pixel is the recommended resolution of MV in the H.264 standard [10]. The frames are interpolated to be 16 times their original size before the ME is performed. The interpolation defined in H.264 includes two stages, interpolating the half-pixel and quarter-pixel sub positions. In the first stage the interpolation is separable, this means the sampling rate in one direction is doubled by inserting zero-valued samples followed by filtering using a 1-D filter h1, [1, -5, 20, 20, -5, 1], and then the process is repeated for the other direction. In the second stage bilinear filtering supported by the integer-pixels is used to generate the interpolated quarter-pixel values. The process is shown in figure 5.1.

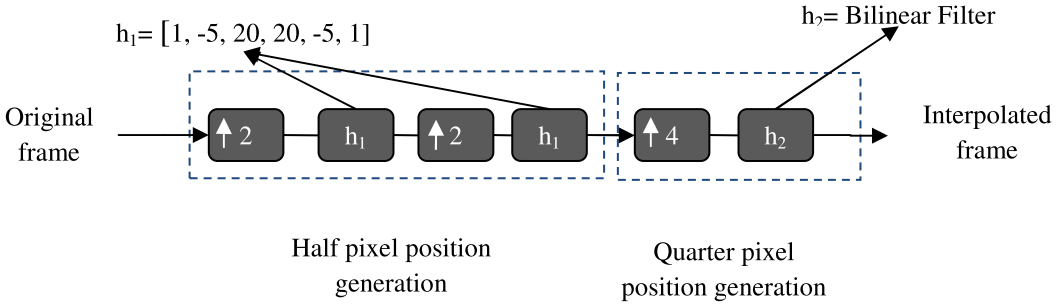

h_1= [1, -5, 20, 20, -5, 1]
h_2= Bilinear Filter
Original frame
↑2
h_1
↑2
h_1
↑4
h_2
Interpolated frame
Half pixel position generation
Quarter pixel position generation

Figure 5.1 - Interpolation process

5.2.3 SATD in video coding

In hybrid video coding approach following the ME that exploits temporal statistical dependencies as described in chapter 2 [8, 10], a transform coding of the prediction residual is performed to utilise spatial statistical dependencies. Figure 5.2 shows the scope of the standard.

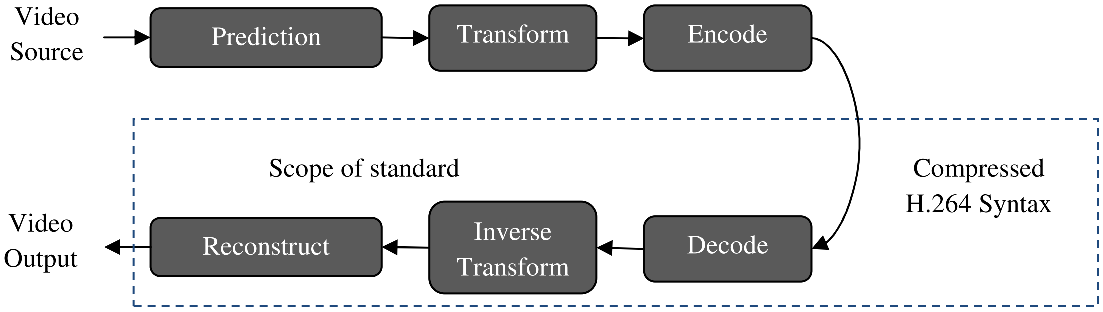


Figure 5.2 – Scope of video coding standardisation [10]

Each colour component of the prediction residual signal is subdivided into smaller 4×4 blocks for the transform coding purposes. Each block is transformed using an integer transform, and the transform coefficients are quantised and encoded using entropy coding methods.

In H.264/AVC, the transformation is applied to 4×4 blocks, and instead of a 4×4 DCT , a separable integer transform with similar properties as a 4×4 DCT is used. The transform matrix is shown in figure 5.3.

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}$$

Figure 5.3 - Integer transform matrix

The Hadamard transform is the simplest orthogonal transform and eliminates spatial redundancies of image therefore it is usually considered as a coarse approximation of DCT. The Hadamard transform matrix is shown in figure 5.4.

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

Figure 5.4 - Hadamard transform matrix

This can be clearly realised when figure 5.3 and figure 5.4 are compared. As a result, ME combined with Hadamard transform is expected to find optimal difference blocks with lower redundancies, which are more suitable for subsequent DCT coding.

5.3 Experimental results

Although the description of section 5.2.3 above holds true when the SATD is simply compared to other error measures metrics, in the H.264 the implementation is far more complicated; as there are two main factors that affect the overall performance. The first one is the Lagrangian cost function and its associate Lagrange multiplier and the second one is the interpolation filters that provide the half-pixel and quarter-pixel search positions [12].

In this section, to illustrate the effects of these factors, two set of experiments have been carried out. Firstly, to demonstrate the effect of the Lagrangian cost function on the SATD, the sub-pixel motion search is disabled, then the SATD performance is compared to the SAD performance in terms of the BDBR percentage difference and the BDPSNR difference in dB [74], the total encoding time differences and the difference in the Distortion Weight (DW) in the Lagrangian cost function. The DW reflects the impact on the number of bits required to encode the residual coefficients and the motion information (more details in equation 5.3 and 5.4). The result is shown in table 5.1.

Secondly, to demonstrate the interpolation effect on the SATD, the sub-pixel motion search is enabled, then the same comparison is preformed. The result is shown in table 5.2.

For clarification purposes, the measures used in the tables are briefly explained here. The minus signs denote PSNR degradation and bitrate savings respectively. Encoding time increase is computed as follows:

$$\Delta T = \frac{T_{SATD} - T_{SAD}}{T_{SAD}} \times 100\% \tag{5.2}$$

The difference in the Distortion DW in the Lagrangian cost function is calculated as follows:

From equation (5.1) DW_{SAD} and DW_{SATD} are calculated using equation 5.3 & 5.4 respectively.

$$DW_{SAD} = \frac{\sum_{ijk} SAD_{ijk}}{\sum_{ijk} SAD_{ijk} + \sum_{ijk} (\lambda_{motion} R_{motion})_{ijk}} \times 100\% \tag{5.3}$$

$$DW_{SATD} = \frac{\sum_{ijk} SATD_{ijk}}{\sum_{ijk} SATD_{ijk} + \sum_{ijk} (\lambda_{motion} R_{motion})_{ijk}} \times 100\% \tag{5.4}$$

Where *i*, *j* are a MB index in frame *k*

Then the difference is calculated using equation 5.5

$$\Delta DW = \frac{DW_{SATD} - DW_{SAD}}{DW_{SAD}} \times 100\% \tag{5.5}$$

Sequence	size	BDPSNR (dB)	BDBR (%)	Time (%)	ΔDW (%)
Akiyo	QCIF	+0.1	-2.1	1053	115.9
	CIF	+0.07	-1.85	1241	144.2
Foreman	QCIF	+0.14	-3.2	710.2	60.2
	CIF	+0.16	-4.16	865.2	63
Mobile	QCIF	+0.12	-1.3	474.9	19.8
	CIF	+0.1	-1.45	623.3	29.7
Stefan	QCIF	+0.08	-1.1	524.5	23.4
	CIF	+0.07	-1.12	665.8	30.6
Silent	QCIF	+0.05	-1.1	900.9	67.1
	CIF	-0.06	-1.6	1055	75.6
Tempete	QCIF	+0.1	-1.5	543.7	28
	CIF	+0.1	-1.72	714.9	75.6
Average		+0.1	-1.85	781	61.1

Table 5.1– Comparison between the SAD and SATD when sub-pixel motion search is disabled

Sequence	size	BDPSNR (dB)	BDBR (%)	Time (%)	ΔDW (%)
Akiyo	QCIF	-0.01	+0.07	999	118
	CIF	+0.01	+0.1	1149	141
Foreman	QCIF	-0.05	+1.05	673.9	56.72
	CIF	-0.01	+0.12	835.4	60.53
Mobile	QCIF	-0.017	+0.2	482.2	20.3
	CIF	0.1	+1.1	661	28.7
Stefan	QCIF	+0.01	+0.01	539.8	23.3
	CIF	0	+0.01	671.3	29.8
Silent	QCIF	-0.01	+0.13	865	66.3
	CIF	-0.1	+0.31	1023	74.3
Tempete	QCIF	+0.01	-0.02	541.3	29.1
	CIF	+0.1	1.65	681	41.1
Average		+0.01	+0.39	760.1	57.4

Table 5.2 – Comparison between the SAD and SATD when sub-pixel motion search is enabled.

Table 5.1 shows the bitrate percentage differences (BDBR) average is -1.85 while the Delta PSNR (BDPSNR) differences average is +0.1 dB. This indicates that although the Hadamard transform outperforms the SAD when sub-pixel ME is enabled, it does not have a significant impact on the RD performance. The reason for that can also be seen from the table where the average DW value when SATD is used is approximately 60% greater than the average DW value when SAD is used. This reduces the contribution of the second part in equation (5.1) and produces more MV bits.

Table 5.2 illustrates the negative effect of the interpolation on the Hadamard transform. From the table it can be seen that when the sub-pixel ME is enabled, although when the SATD is used the average total encoding time is increased by 760%, the RD performance is degraded.

The Hadamard transform aims to match frequencies instead of pixels to get a better performance in the transform/quantisation process by reducing the coefficients bits. However, the above experiments shows that, although the Hadamard transform successfully reduces the coefficients bits significantly, in some cases the Hadamard transform does not result in finding the true motion which makes the interpolation process ineffective and affects other MBs due to the prediction.

Further investigations have been carried out to examine the SATD performance against the SAD performance in sub-pixel search. The results of these investigations indicated that the Hadamard transform outperform the SAD significantly in the sub-pixel search because the search positions are limited to 9 positions (In full-pixel the number of positions = (2×search_range+1)×(2×search_range+1)) which limit the MV range and increase the significance of the distortion term in equation 5.1. Furthermore, the increase in the encoding time can be tolerated.

5.4 Summary

In video coding, the most commonly used ME distortion metrics are predominantly based on the SAD and the SATD. The latter is used since ultimately the transformed coefficients are coded, better estimation of the cost can be achieved by estimating the effect of the DCT with a 4×4 Hadamard transform. Although these advantages are well known and the Hadamard transform is implemented in various parts of the ME and MC processes of the standard, no research has been carried out to investigate the effect of the λ selection and the interpolation on the SATD. The reason for this is the extensive computations required to compute the SATD; which involves subtraction, addition, shift and absolute value operations. However, if the ME is improved to accommodate the use of the SATD in the full-pixel motion search, in addition to enhancing the effect of the DCT, significant RD enhancement may be achieved in wide range of applications. Particularly, in hardware applications when applying the same distortion metric at different resolutions is essential.

In this chapter an in-depth study that looked into the exploitation of SAD and SATD in the ME process of the standard was carried out and concluded. This study is a novel contribution to the body of knowledge as it has a significant result that can lead to a significant improvement in future video coding standards.

In the next chapter another effect of the H.264/AVC is investigated and utilised in the development of an efficient fast sub-pixel ME algorithm. The newly developed method terminate sub-pixel ME adaptively, which leads to a signification time saving in the SATD calculations.

CHAPTER 6
SELECTIVE APPLICATION OF SUB-PIXEL MOTION ESTIMATION AND HADAMARD TRANSFORM IN H.264/AVC

6.1 Introduction

The computational efficiency of full and sub-pixel ME is a key implementation issue for real-time video encoding. This is because both full and sub-pixel ME are always computational bottlenecks in any hybrid video coding system [11]. Computational analysis was carried in chapter 2 and showed that full and sub-pixel ME is accountable for 80% of the overall encoding time.

In the previous chapter, the importance of sub-pixel ME has been outlined and the H.264 interpolation methods for generating sub-pixel positions were discussed. Furthermore, the interpolation effect on different distortion metrics used in the ME process was investigated. In this chapter a drawback of the H.264 recommended sub-pixel ME is utilised and a novel fast sub-pixel ME algorithm is proposed [75, 76]. The correlation between a MB and its enclosed partitions is exploited to terminate efficiently unnecessary sub-pixel ME. This has been carried out in order to reduce the computational cost of the overall ME process.

The chapter starts with a survey of recently published fast sub-pixel ME algorithms. Then the proposed algorithm is explained in details followed by a comprehensive list of experiments and a discussion.

6.2 Fast Sub-Pixel Motion Estimation Algorithms

6.2.1 Overview

Different fast sub-pixel ME algorithms [77- 80] have been proposed, and a number of them have been adopted by the JM reference software [30]. The common idea is to simplify the search pattern by applying refined prediction algorithms, and improved adaptive threshold schemes to terminate unnecessary search positions.

A few algorithms decrease the computational complexity by reducing the number of search points such as the algorithms in [81, 82]. These algorithms have an irregular search pattern, which makes them unsuitable for dedicated hardware implementation.

In [83] and [84] an interpolation free sub-pixel ME algorithms was proposed, these methods use the ME cost of neighbouring integer-pixel positions and a parabolic model to form the distortion surface of sub-pixel resolution positions, from which the best-match sub-pixel position could be found without image interpolation. Although those algorithms provide reasonable accuracy their computational time saving is very limited.

Another scheme for candidate reduction in sub-pixel ME has been proposed in [85]. In this, the ME cost of neighbouring integer-pixel points is used to generate the error surface, then, the sub-pixel candidate positions is checked selectively. In contrast to the algorithms proposed in [83] and [84] this algorithm achieves a clear time saving, however this has been accomplished whilst increasing the bitrate.

Also the utilisation of mathematical models for matching error either to reduce the number of fractional-pixel search points [79, 86] or to directly predict the best matching fractional-pixel location [87, 88] were proposed.

6.2.2 Recent Algorithms

In [89] a low-complexity algorithm for fractional-pixel ME was proposed. The algorithm uses a mathematical model to approximate the matching error at fractional-pixel locations. This scheme is shown in figure 6.1 as a flowchart.

This algorithm has two drawbacks; firstly it requires the estimation of the matching errors for external points to fit the model. Secondly, it does not take into consideration the video contents which leads to performance degradation for fast video sequences where the Full Search (FS) sub-pixel ME is more efficient.

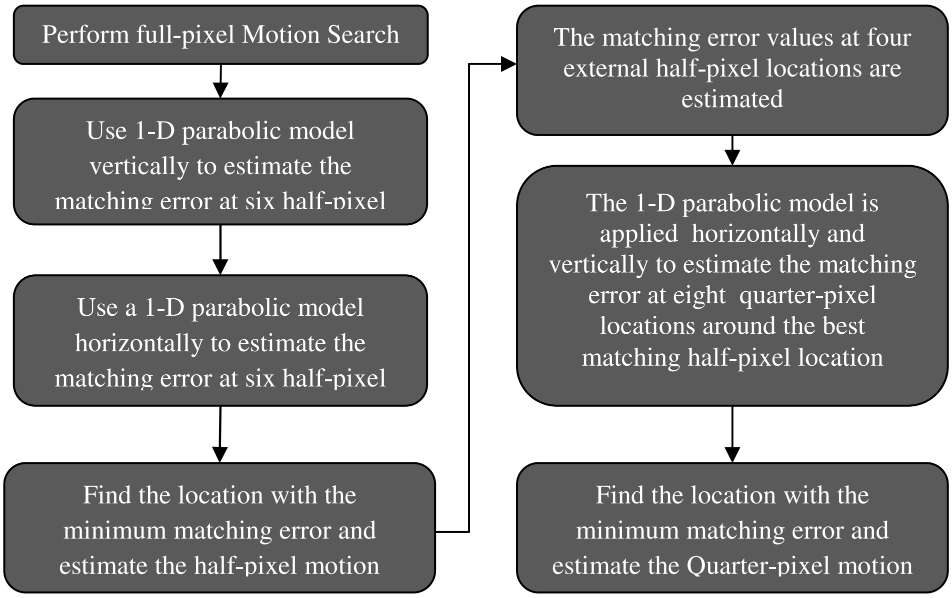

Perform full-pixel Motion Search
Use 1-D parabolic model vertically to estimate the matching error at six half-pixel
Use a 1-D parabolic model horizontally to estimate the matching error at six half-pixel
Find the location with the minimum matching error and estimate the half-pixel motion
The matching error values at four external half-pixel locations are estimated
The 1-D parabolic model is applied horizontally and vertically to estimate the matching error at eight quarter-pixel locations around the best matching half-pixel location
Find the location with the minimum matching error and estimate the Quarter-pixel motion

Figure 6.1 – Sayed's fast Sub-pixel ME algorithm [89]

In [90] an algorithm that uses the statistical information featuring the motion activities of the blocks in the previous frame has been proposed. The characteristics of the motion activities of the blocks in the current frame are predicted using this statistical information. The algorithm is shown in figure 6.2 as a flowchart.

If the MV is not equal to the best predicted MV, the current block is classified into the special block group. Otherwise, it is an ordinary block. Then, sub-pixel refinement is carried out surrounding the best integer-pixel point. The sub-pixel refinement procedure is done in two parts: half-pixel refinement and quarter-pixel refinement. For half and quarter-pixel refinements, the 4 nearest neighbours points are checked, then the 2 far neighbour points beside the best near neighbour point are checked. Before quarter-pixel refinement, decision on early termination is made. The conditions for early termination are:

1. The current block is a prejudged ordinary block.

2. The MV of its co-located block is a zero vector,

3. The MV of the current block is a zero vector after half-pixel refinement.

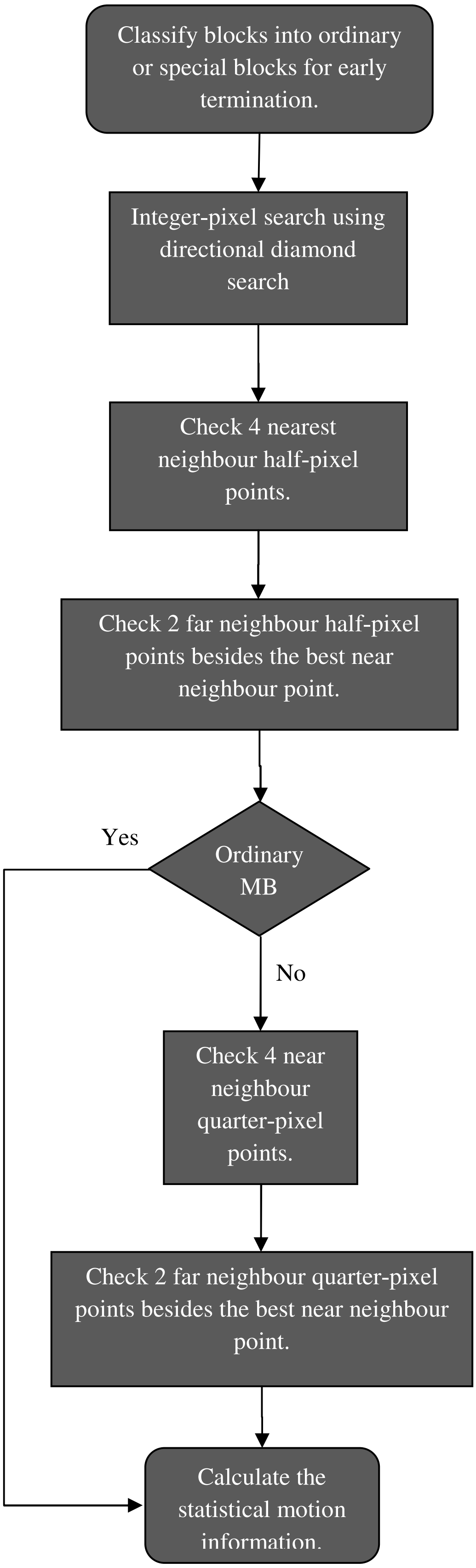

Classify blocks into ordinary or special blocks for early termination.
Integer-pixel search using directional diamond search
Check 4 nearest neighbour half-pixel points.
Check 2 far neighbour half-pixel points besides the best near neighbour point.
Ordinary MB
Yes
No
Check 4 near neighbour quarter-pixel points.
Check 2 far neighbour quarter-pixel points besides the best near neighbour point.
Calculate the statistical motion information.

Figure 6.2 – Zhang’s fast sub-pixel ME algorithm [90]

From figure 6.2 it can be seen the algorithm requires an extra step to calculate MB's statistical information which limits the time saving to some extent. Furthermore this information has to be stored for at least the time that takes to encode a frame which increases the memory requirements.

In [91] a novel scheme for sub-pixel ME using a model free kernel method utilising the result of the integer-pixel SAD distribution applied to an H.264 encoder has been presented. The algorithm is shown in figure 6.3 as a flowchart.

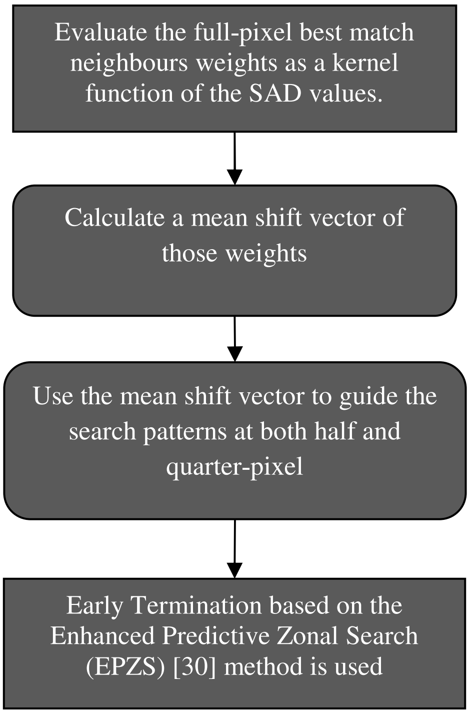

Evaluate the full-pixel best match neighbours weights as a kernel function of the SAD values.
Calculate a mean shift vector of those weights
Use the mean shift vector to guide the search patterns at both half and quarter-pixel
Early Termination based on the Enhanced Predictive Zonal Search (EPZS) [30] method is used

Figure 6.3 – Hill's Kernel Based Sub-pixel ME algorithm [91]

The SAD values of the integer-pixel ME surrounding the best match is weighted, then a mean shift factor of those weights is evaluated and used to direct the search pattern in a hierarchal approach.

6.3 Proposed Scheme

Video objects in a frame are not always deformed or divided with respect to a small temporal window in the past. They may be motionless as a part of a background or just incur small translational changes, especially for slow motion video sequences or for the slow motion segments of fast video sequences. In these situations, the ME process might find a best match position during the integer-pixel motion search which does not change after the subsequent fractional-pixel motion search. Furthermore, if the integer-pixel ME best match for a bigger block size does not change during fractional-pixel motion search, it is highly likely that this integer-pixel ME result for the partitions of this block will also not change during the fractional-pixel motion search. This likelihood depends on the difference in block sizes as demonstrated below. The above observations are shown in the tables 6.1-6.3 within a probabilistic framework.

These observations which is supported by the probabilistic framework is utilised to develop a novel early termination sub-pixel ME scheme. The complexity of fractional-pixel ME is decreased by effectively applying a two step algorithm. Firstly the 16×16 MB fractional-pixel ME best match is examined, derived from the outcome the fractional-pixel motion search for 16×8 and 8×16 MB partitions is conditionally eliminated. Likewise, in the second step, the 8×8 MB partitions fractional-pixel ME best matches are examined and derived from the outcome the fractional-pixel motion search for 8×4, 4×8 and 4×4 MB partitions is eliminated.

Table 6.1 shows the probability of the 16×8 and 8×16 MB partitions having the same best match in integer and fractional-pixel ME, given that the 16×16 MB has the same best match in integer and fractional-pixel ME. This has been called (P_1).

Table 6.2 shows the probability of 8×8, 8×4, 4×8 and 4×4 blocks having the same best match in integer and fractional-pixel ME, given that the 16×16 MB has the same best match in integer and fractional-pixel ME. This has been called (P_2).

Tables 6.3 shows the probability of 8×4, 4×8, 8×4 and 4×4 blocks partitions having the same best match in integer and fractional-pixel ME, given that the 8×8 blocks have the same best match in integer and fractional-pixel ME. This has been called (P_3).

Sequence	Size	Quantisation Parameter			
		14	22	30	38
Akiyo	QCIF	87.83	85.51	83.6	88.48
	CIF	84.31	85.81	88.95	93.4
Foreman	QCIF	69.66	65.43	59.19	58.14
	CIF	66.86	63.99	60.05	72.91
Mobile	QCIF	76.95	75.92	73.33	56.19
	CIF	74.92	72.57	66.7	59.79
Stefan	QCIF	80.25	78.4	77.13	71.09
	CIF	65.99	67.66	74.4	74.79
Silent	QCIF	92.28	89.02	77.58	74.61
	CIF	91.72	90.83	80.83	85.21
Tempete	QCIF	80.98	78.44	73.7	65.43
	CIF	74	69.13	60.89	71.93
Average		78.81	76.89	73.029	72.66

Table 6.1 – Evaluation of P_1 expressed in terms of percentage

Sequence	Size	Quantisation Parameter			
		14	22	30	38
Akiyo	QCIF	78.61	81.53	82.56	91.99
	CIF	68.57	83.49	90.21	95.22
Foreman	QCIF	50.69	48.76	47.88	58.67
	CIF	48.08	50.52	54.97	74.21
Mobile	QCIF	62.01	59.82	55.56	43.49
	CIF	60.39	58.42	53.4	48.38
Stefan	QCIF	65.33	68.13	67.15	62.97
	CIF	51.89	59.86	66.92	70.84
Silent	QCIF	84.84	80.24	71.39	82.03
	CIF	85.22	82.3	75.76	89.18
Tempete	QCIF	61.7	60.39	56.07	56.77
	CIF	58.03	55.48	51.51	62.94
Average		64.61	65.75	64.45	69.72

Table 6.2 – Evaluation of P_2 expressed in terms of percentage

Sequence	Size	Quantisation Parameter			
		14	22	30	38
Akiyo	QCIF	83.07	85.64	86.65	92.48
	CIF	73.37	86.43	91.88	95.17
Foreman	QCIF	61.11	55.59	61	74.32
	CIF	55.5	57.32	67.46	83.47
Mobile	QCIF	60.8	59.66	56.66	48.78
	CIF	61.42	60.8	58.28	54.58
Stefan	QCIF	68.72	72.07	72.71	70.17
	CIF	55.92	69.09	76.36	78.28
Silent	QCIF	89.74	84.41	78.17	85.42
	CIF	88.68	83.82	81.39	91.13
Tempete	QCIF	67.94	66.27	60.78	64.78
	CIF	60.51	58.93	60.24	63.25
Average		68.9	70	71	75.1

Table 6.3 – Evaluation of P_3 expressed in terms of percentage

From the tables, it can be seen that the conditional probabilities are reasonably high only when the macroblock/block and their partitions do not have a significant size difference. For example, it cannot safely be said that the 8×4, 4×8 and 4×4 partitions would find the same best match in the integer and fractional-pixel motion search, given that the enclosing 16×16 MB does so. In this case the difference in size is significant, since the 16×16 MB is 8, 8 and 16 times greater with respect to the aforementioned block sizes.

Further conditional probabilities have been evaluated to examine the effects of P_1 and P_3 on the chosen block size. Moreover, given that a 16×16 MB has the same best match in integer and fractional-pixel-ME but the 16×8 and 8×16 MB partitions do not have the same best match in integer and fractional-pixel ME, the probabilities of best mode being 16×8 or 8×16 have been measured. The result of this evaluation is shown in table 6.4. This has been called (P_4).

Sequence	Size	Quantisation Parameter			
		14	22	30	38
Akiyo	QCIF	2.2	2.97	2.03	1.08
	CIF	4.2	5.1	3.47	1.18
Foreman	QCIF	6.63	10.27	9.6	5.2
	CIF	6.58	10.7	11.91	5.99
Mobile	QCIF	7.71	6.73	5.45	9.32
	CIF	8.36	6.77	5.16	8.56
Stefan	QCIF	7.95	8.64	7.85	10.31
	CIF	5.33	5.87	5.07	7.64
Silent	QCIF	2.59	4.11	4.3	2.58
	CIF	3.22	5.21	5.17	3.14
Tempete	QCIF	7.31	6.94	8.74	8.56
	CIF	5.26	4.92	6.54	7.25
Average		6.52	6.52	6.27	5.9

Table 6.4 - Evaluation of P_4 expressed in terms of percentage

Furthermore, given that an 8×8 MB has the same best match in integer and fractional-pixel-ME but the 8×4, 4×8 and 4×4 MB partitions do not have the same best match in integer and fractional-pixel ME, the probabilities of best mode being 8×4, 4×8 or 4×4 have been measured, P_5. The result of this evaluation is shown in table 6.5.

From table 6.4 and 6.5, it can be seen that when a 16×16 MB has the same best match in integer and fractional-pixel ME it is very unlikely that the best mode for that MB is 8×16 or 16×8 (an average of 6.3%). Similarly, when a 8×8 partition has the same best match in integer and fractional-pixel ME it is very unlikely that the best mode for that partition is 8×4, 4×8 or 4×4 (an average of 11.6%).

Sequence	Size	Quantisation Parameter			
		14	22	30	38
Akiyo	QCIF	13.07	15.02	11.24	3.25
	CIF	12.5	6.71	3.34	0.98
Foreman	QCIF	16.45	14.76	10.68	5.93
	CIF	14.68	7.69	5.06	1.27
Mobile	QCIF	18.51	18.47	19.74	17.3
	CIF	15.94	15.87	14.69	7.67
Stefan	QCIF	16.84	17.07	17.82	16.45
	CIF	12.35	10.98	11.09	4.74
Silent	QCIF	17.55	15.5	13.2	7.8
	CIF	15.06	8.9	3.93	1.1
Tempete	QCIF	14.99	14.36	15.16	11.28
	CIF	14.05	13.09	9.79	3.66
Average		15.17	13.2	11.3	6.79

Table 6.5 - Evaluation of P_5 expressed in terms of percentage

Based on the above insights, the following scheme has been developed:

1. If a 16×16 MB finds the same best match in the integer and fractional-pixel-motion-searches, then the fractional-pixel motion search is disabled for all the enclosed 16×8 and 8×16 blocks. Thus, all the fractional-pixel search, SAD and Hadamard transform calculations have been saved for these blocks, otherwise, the fractional-pixel motion search is performed.

2. Similarly, if the 8×8 block partitions of the 16×16 MB find the same best match in the integer and fractional-pixel motion searches, the fractional-pixel motion search is disabled for all the enclosed 8×4, 4×8 and 4×4 blocks. Otherwise, the fractional-pixel motion search is performed.

The algorithm differs from the previously mentioned methods in two aspects.

1. It uses the correlation between the interpolation effect on the MB and its partitions to completely eliminate the fractional-pixel-ME.

2. The proposed algorithm is adaptive and can be applied to any combination of integer and fractional-pixel-ME schemes.

This algorithm utilises the statistical information, which features the motion activities of the block and its partitions. With this prior knowledge, the fractional-pixel ME is adaptively terminated.

The proposed algorithm is shown on figure 6.4 as a flowchart:

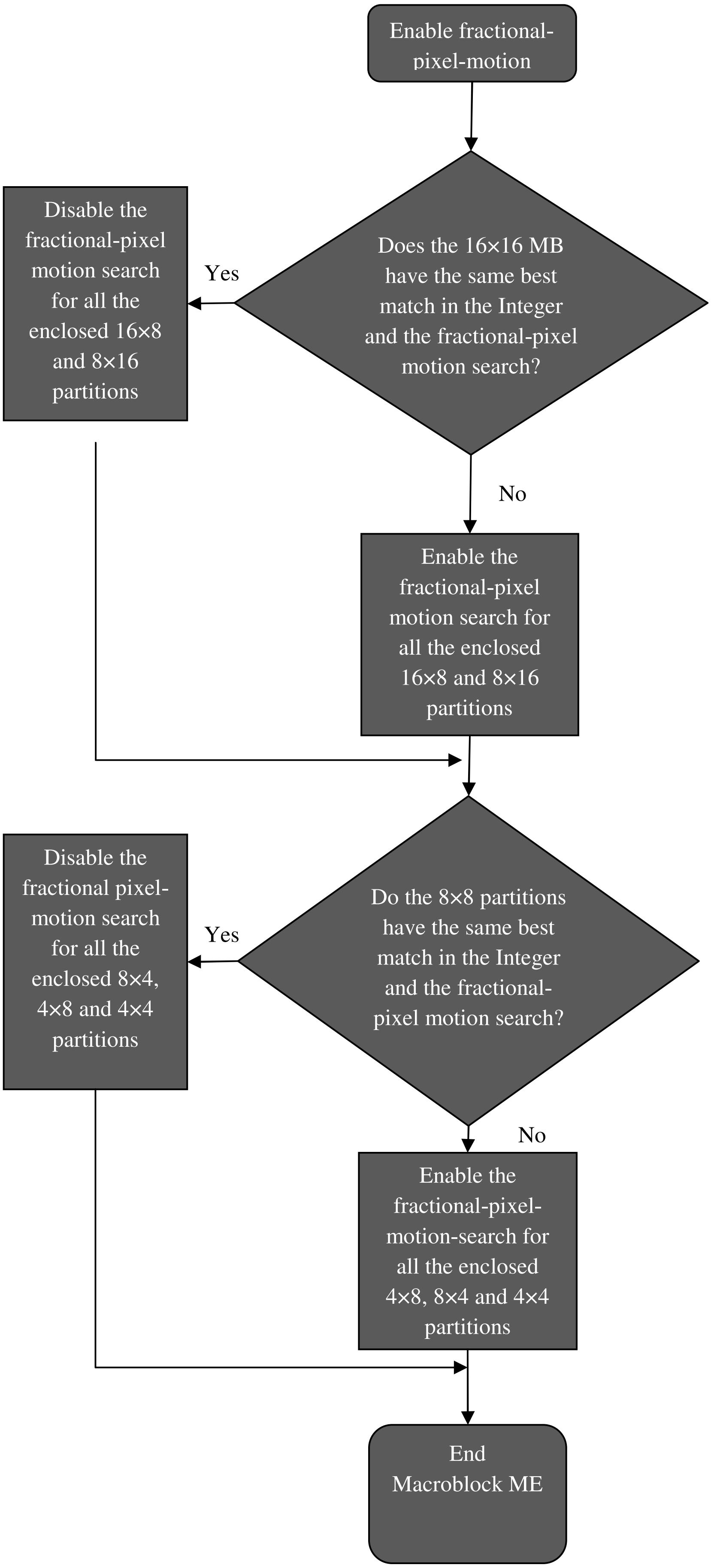

Enable fractional-pixel-motion
Does the 16×16 MB have the same best match in the Integer and the fractional-pixel motion search?
Yes
Disable the fractional-pixel motion search for all the enclosed 16×8 and 8×16 partitions
No
Enable the fractional-pixel motion search for all the enclosed 16×8 and 8×16 partitions
Do the 8×8 partitions have the same best match in the Integer and the fractional-pixel motion search?
Yes
Disable the fractional pixel-motion search for all the enclosed 8×4, 4×8 and 4×4 partitions
No
Enable the fractional-pixel-motion-search for all the enclosed 4×8, 8×4 and 4×4 partitions
End Macroblock ME

Figure 6.4 – The proposed algorithm.

6.4 Experimental Results

6.4.1 Overview

To assess the proposed algorithm, a comprehensive set of experiments were conducted using a variety of video sequences with different motion characteristics. Details of those sequences and the experimental setup conditions can be found in chapter 3. The following sections show the percentage cycle savings, the BDBR percentage differences and BDPSNR differences (in dB) between the H.264/AVC reference software and recently reported against the proposed algorithm. While the percentage cycle savings show the algorithm effectiveness when the overall encoding time is considered, in addition, a percentage of the saved search positions for sub-pixel ME is shown to emphasis the algorithm effect on sub-pixel ME.

The exact configuration settings for these experiments is shown table 6.6

Parameter	Value		Parameter	Value
Profile	100 (Main)		**YUV Format**	YUV 4:2:0
Level IDC	40		**B-Frame**	Not Used
Entropy Coding	CABAC		**Frame Skip**	0
References	1,3 and 5		Search range	32
ME Metric Level 0	SAD		**ME Metric Level 1&2**	Hadamard SAD

Table 6.6 - Encoder experiments conditions

6.4.2 Comparison against the H.264/AVC reference software

Initially the proposed algorithm has been compared against the H.264/AVC reference software [30] when number of different full and sub-pixel ME schemes have been used. This has been carried out to demonstrate that the proposed algorithm is adaptable to different ME schemes and suitable for a variety of encoding settings.

In table 6.7 a comparison between the proposed algorithm and the JM software when Fast Full Search (FFS) [30] is used as the full-pixel ME Scheme, and Full Search (FS) [29] is used as the sub-pixel ME scheme is presented. The FFS scheme has been described in chapter 2. The FFS scheme is a fast software implementation of the FS.

Full pixel **Motion Estimation Scheme**				**Sub-pixel** **Motion Estimation Scheme**			
FFS				FS			
Sequence	Size	BDPSNR*(dB)*	BDBR *(%)*	Proposed T (s)	JM T (s)	Cycles *(%)*	Search point Saving *(%)*
Akiyo	QCIF	+0.02	-0.45	200.4	214.33	6.5	62.8
	CIF	-0.04	-0.12	786.45	586.7	8.2	62.13
Foreman	QCIF	-0.05	+1.2	218.87	224.7	2.45	16.19
	CIF	-0.01	+0.28	860.9	885.4	2.77	15.73
Mobile	QCIF	-0.03	+0.43	245.97	250.09	1.98	12.84
	CIF	-0.01	+0.12	913.47	929.09	1.68	7.21
Stefan	QCIF	-0.03	+0.51	372.12	379.26	1.88	12.43
	CIF	-0.03	+0.51	887.55	902.7	1.87	11.35
Silent	QCIF	-0.02	+0.48	288.6	307.75	6.2	55.86
	CIF	-0.02	+0.55	816.35	868.55	6.01	56.42
Tempete	QCIF	-0.01	+0.2	362.14	369.15	1.9	11.2
	CIF	-0.01	-0.01	893.56	906.99	1.15	7.59

Table 6.7– Comparison between the Proposed Algorithm and JM when FFS used as the Full-pixel ME scheme and FS used as the Sub-pixel ME scheme

In table 6.8 a comparison between the proposed algorithm and the JM software when Unsymmetrical-cross Multihexagon-grid Search (UMHEXS) [77] is used as the full and sub-pixel ME is presented.

Full pixel **Motion Estimation Scheme**				**Sub-pixel** **Motion Estimation Scheme**			
UMHEXS				UMHEXS			
Sequence	Size	BDPSNR*(dB)*	BDBR *(%)*	Proposed T (s)	JM T (s)	Cycles *(%)*	Search point Saving *(%)*
Akiyo	QCIF	-0.01	+0.34	81.02	94.56	14.32	67.07
	CIF	-0.01	+0.43	307.66	366.527	16.06	67.76
Foreman	QCIF	-0.02	+0.49	117.471	121.03	2.94	14.48
	CIF	-0.03	+0.72	437.07	451.39	3.17	14.73
Mobile	QCIF	-0.05	+0.51	178.32	182.13	2.09	12.31
	CIF	-0.02	+0.31	569.87	576.09	1.08	6.95
Stefan	QCIF	-0.03	+0.52	153.07	156.52	2.2	11.67
	CIF	-0.01	+0.12	522	531.03	1.7	10.8
Silent	QCIF	-0.01	+0.1	95.63	110.12	13.16	56.42
	CIF	-0.01	+0.37	362.48	410.92	11.79	57.3
Tempete	QCIF	-0.02	+0.25	158.10	165.66	4.5	11.3
	CIF	-0.01	+0.16	523.8	529.541	1.08	7.7

Table 6.8– Comparison between the Proposed Algorithm and the H.264/ AVC when UMHEXS is used as the Full and Sub-pixel ME Scheme

In table 6.9 a comparison between the proposed algorithm and the JM software when UMHEXS is used as the full-pixel ME Scheme, and FS is used as the sub-pixel ME scheme is presented.

Full pixel Motion Estimation Scheme					Sub-pixel Motion Estimation Scheme		
UMHEXS					FS		
Sequence	Size	BDPSNR*(dB)*	BDBR *(%)*	Proposed T (s)	JM T (s)	Cycles *(%)*	Search point Saving (%)
Akiyo	QCIF	0.0	0.0	95.03	111.6	14.85	63.22
	CIF	-0.02	+ 0.49	363.44	432.10	15.89	62.82
Foreman	QCIF	-0.02	+0.67	132.17	136.262	3	13.5
	CIF	-0.02	+0.6	496.67	513.24	3.23	13.94
Mobile	QCIF	-0.04	+0.35	192.19	196.11	2	12.2
	CIF	-0.01	+0.17	625.87	633.53	1.21	6.57
Stefan	QCIF	-0.04	+0.01	167.41	171	2.1	11.14
	CIF	-0.01	+0.23	578.34	590.18	2	10.39
Silent	QCIF	-0.03	+0.48	110.13	145.58	11.6	56.15
	CIF	-0.02	+0.47	419.46	480.2	12.65	57.07
Tempete	QCIF	-0.03	+0.33	172.52	175.89	1.92	11.11
	CIF	0.0	+0.06	580.09	588.328	1.4	7.35

Table 6.9– Comparison between the proposed algorithm and the H.264/ AVC when UMHEXS is used as the full-pixel ME scheme and FS is used as the sub- pixel ME Scheme

In table 6.10 a comparison between the proposed algorithm and the JM software when Enhanced Predictive Zonal Search (EPZS) [78] is used as the full and sub-pixel ME is presented.

Full pixel Motion Estimation Scheme					Sub-pixel Motion Estimation Scheme		
EPZS					EPZS		
Sequence	Size	BDPSNR*(dB)*	BDBR *(%)*	Proposed T (s)	JM T (s)	Cycles *(%)*	Search point Saving (%)
Akiyo	QCIF	-0.01	+0.24	80.478	90.35	10.9	63.2
	CIF	-0.01	+0.25	308.42	346.69	11.04	65.4
Foreman	QCIF	-0.03	+0.84	119.90	124.38	3.61	18.45
	CIF	-0.04	+0.81	469.47	485.14	3.23	17.76
Mobile	QCIF	-0.03	+0.39	150.12	153.23	2.03	13.17
	CIF	-0.01	+0.16	547.31	554.86	1.36	7.44
Stefan	QCIF	-0.13	+0.2	137.07	139.96	2.07	12.47
	CIF	-0.01	+0.2	505.84	515.59	1.89	11.55
Silent	QCIF	-0.02	+0.43	95.37	104.71	8.92	57.56
	CIF	-0.02	+0.46	374.85	413.6	9.37	58.75
Tempete	QCIF	-0.03	+0.37	138.09	141.11	2.14	11.53
	CIF	-0.01	+0.02	520.1	527.17	1.34	7.97

Table 6.10– Comparison between the Proposed Algorithm and the H.264/ AVC when EPZS is used as the Full and Sub-pixel ME Scheme

In table 6.11 a comparison between the proposed algorithm and the JM software when EPZS is used as the full-pixel ME Scheme and FS is used as the sub-pixel ME scheme is presented.

Full pixel Motion Estimation Scheme					Sub-pixel Motion Estimation Scheme		
EPZS					FS		
Sequence	Size	BDPSNR*(dB)*	BDBR *(%)*	Proposed T (s)	JM T (s)	Cycles *(%)*	Search point Saving (%)
Akiyo	QCIF	-0.01	+0.11	95.9	112.7	14.9	63.18
	CIF	-0.02	+0.49	374.04	440.78	15.14	62.95
Foreman	QCIF	-0.01	+0.24	129.46	138.75	6.7	14.64
	CIF	-0.01	+0.29	512.275	530.03	3.35	14.84
Mobile	QCIF	-0.03	+0.28	159.52	161.62	1.3	13.02
	CIF	-0.02	+0.38	579.99	587.56	1.29	6.72
Stefan	QCIF	-0.02	+0.4	145.93	149.52	2.4	12.74
	CIF	-0.01	+0.24	543.06	555.27	2.2	10.91
Silent	QCIF	+0.01	-0.14	109.08	123.55	11.71	56.5
	CIF	-0.02	+0.43	427.542	489.18	12.6	57.71
Tempete	QCIF	-0.01	+0.29	149.74	151.93	1.44	11.31
	CIF	-0.01	+0.18	552.3	569.67	3.05	7.48

Table 6.11– Comparison between the proposed algorithm and the H.264/ AVC when EPZS is used as the full-pixel ME scheme and FS is used as the sub- pixel ME Scheme

The above tables show that the bitrate percentage differences (BDBR) are in the range of [-0.5, 1.2], while the Delta PSNR (BDPSNR) differences are in the range of [-0.04, 0.02]. The negative signs denote PSNR degradation and bitrate savings respectively.
This clearly shows that the proposed algorithm has very similar RD performance to the H.264/AVC reference software. Furthermore, percentage search point savings up to 67% and percentage cycle savings up to 16% are observed. It also can be seen that the reduction in the CPU cycles depends on the characteristics of the image sequences. For a slow image sequence with a simple background, the reduction is much more significant than for a fast image sequences or sequences with a more complex background.

6.4.3 Comparison against recently proposed algorithm

Further experiments have been carried out to compare the proposed scheme to recently published work [89, 90].

The result of these experiments is shown in table 6.12 and 6.13. From the two tables it can be seen the proposed algorithm time saving is better than two algorithm whilst maintain the same RD performance.

Sequence	Size	BDPSNR*(dB)*	BDBR *(%)*	Cycles *(%)*
Akiyo	QCIF	+0.01	-0.1	2.19
	CIF	+0.05	-0.21	2.56
Foreman	QCIF	+0.11	+0.09	1.1
	CIF	-0.03	-0.32	1.36
Mobile	QCIF	+0.07	-1.1	1.08
	CIF	-0.16	-1.2	0.78
Stefan	QCIF	-0.12	+0.11	0.65
	CIF	+0.01	-0.39	0.9
Silent	QCIF	+0.05	-0.04	3.2
	CIF	+0.07	+0.12	2.61
Tempete	QCIF	-0.03	-0.75	0.97
	CIF	-0.04	-0.76	1.4

Table 6.12 – Comparison between the proposed algorithm and Sayed's algorithm [89].

Sequence	Size	BDPSNR*(dB)*	BDBR *(%)*	Cycles *(%)*
Akiyo	QCIF	0.04	-0.31	1.36
	CIF	0.12	-0.29	1.76
Foreman	QCIF	0.03	-0.04	2.22
	CIF	0.02	-0.49	2.01
Mobile	QCIF	0.16	-0.07	2.57
	CIF	0.07	-0.12	2.49
Stefan	QCIF	0.05	-1.11	1.93
	CIF	0.11	-1.09	1.98
Silent	QCIF	0.05	+0.27	1.4
	CIF	0.08	-0.57	1.56
Tempete	QCIF	0.03	-0.81	2.12
	CIF	0.06	-0.11	2.32

Table 6.13 – Comparison between the proposed algorithm and Zhang's algorithm [90]

6.5 Summary

The H.264 encoding method has been complicated by the development of new coding tools. Amongst those tools is the quarter sample accurate MC, while this enhance the RD performance as discussed in the previous chapter, it requires the implementation of complex interpolation filters and increases the ME complexity. In this chapter a new complexity reduction sub-pixel ME algorithm for an H.264/AVC encoder was proposed. The objective of the algorithm is to match the RD performance of the un-optimised encoder while significantly reducing the computational complexity compared to the same. The new algorithm utilises the correlation between the MB and its enclosed partitions to effectively terminate sub-pixel motion search adaptively. The scheme is divided into two steps; in the first step the sub-pixel ME for the 8×16 and 16×8 block is eliminated based on the 16×16 sub-pixel ME results, in the second step sub-pixel ME for the 8×4, 4×8 and 4×4 blocks is eliminated based on the 8×8 sub-pixel ME results. For RD performance similar to the standard, the proposed scheme can reduce up to 67% of sub-pixel search points resulting in significant time/cycle savings as compared to the standard reference software. Unlike most algorithms available in literature, the performance of the complexity reduction algorithm does not depend on empirically obtained thresholds. This algorithm automatically adapts to different sequence statistics without the need for tuning of any thresholds.

For high/moderate-high activity sequences such as Foreman and mobile, significant reductions in computational complexity can be achieved, more than 19% with only negligible loss in RD performance. For low activity sequences such as Akiyo and Silent, the RD performance is better than the un-optimised encoder with complexity reductions of more than 50% obtained. The algorithm is very relevant to low complexity video coding systems.

The main drawback of the proposed approach is that; the reduction in the CPU cycles is limited when fast sequences or sequences with a more complex background is coded. However, this directs the algorithm applications to video telephony and video conferencing where Bandwidth and quality of service are still limited while their applications are widely spreading.

This algorithm forms the foundation for further research that resulted in the development of complexity control and complexity management algorithms, described in chapters 7 and 8. This chapter has contributed to the body of knowledge by two publications [75] and [76].

CHAPTER 7
LOW COMPLEXITY HIERARCHICAL PREDICTION ALGORITHM FOR H.264/SVC

7.1 Introduction

The objective of SVC is to enable the generation of a unique bitstream that can adapt to various bitrates, transmission channels and display capabilities. The scalability is categorised into temporal, spatial, and quality. For temporal and spatial scalabilities, subsets of the bit stream (referred to as sub-streams) represent the source content with a reduced frame rate (temporal resolution) or picture size (spatial resolution), respectively. With quality scalability, the sub-stream provides the same spatial and temporal resolutions as the complete bit stream, but with a lower signal-to-noise ratio (SNR) [14].

In order to improve coding efficiency, the SVC scheme incorporates inter-layer prediction mechanisms to complement the H.264/AVC very refined Motion Estimation (ME) and mode decision processes. However, this further increases the overall encoding complexity of the scalable coding standard.

In the movement toward the Next Generation Video Coding (NGVC) H.265 the concept of adaptive interpolation filters that are specifically adapted to statistics of the current image are introduced in [92], and motion compensated prediction with 1/8-pel displacement vector resolution is discussed in [93]. However, little research has been carried out to investigate and define situations when the interpolation process is unnecessary. In this chapter the algorithm proposed in the previous chapter is extended to a Group of Pictures (GOP) in H.264/SVC. Moreover, the algorithm has been utilised in the mode selection process of different scalability layers to reduce the computational complexity [94, 95].

The chapter is organised as follows. Section 7.2 gives a brief overview of the hierarchical prediction proposed in the H.264/SVC. Section 7.3 discuses some related work in the field. Section 7.4 describes the proposed mode decision algorithm, while section 7.5 presents details and discussion of a comprehensive list of comparative experimental results. Section 7.6 summarises the chapter.

7.2 Scalable Video Coding

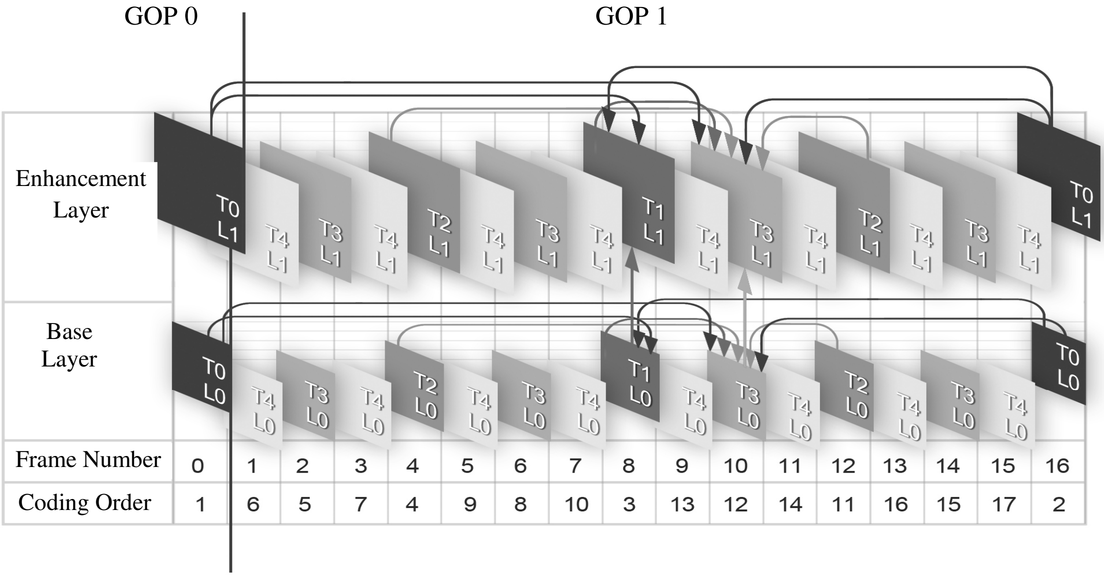
7.2.1 Overview

The following subsection outlines the different type of scalability and the main tools of SVC.

7.2.2 Group of Pictures Hierarchical Prediction

In H.264/SVC a Group of Pictures (GOP) is an encoding of a contiguous subset of frames from a video sequence; each GOP consists of all frames between two successive frames at the lowest temporal resolution, plus the second of the temporal base layer frames (the first is considered to belong to the previous GOP).

All information required to decode any one frame from the GOP is contained within it. Using the GOP concept, SNR gains of more than 1 dB can be obtained for medium bitrates when compared to the widely used IBBP coding structure [14].



Frame Number	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Coding Order	1	6	5	7	4	9	8	10	3	13	12	14	11	16	15	17	2

Figure 7.1 - An example of a 2 layer GOP of size 16 for enabling temporal and spatial scalability

Figure 7.1 shows an example of a 2 layer GOP of size 16. This example shows a dyadic coding structure as this is the most commonly used structure on account of its demonstrated coding efficiency.

The figure illustrates a case when the base and the enhancement layers have the same frame rate, however the H.264/SVC supports non-dyadic hierarchical prediction structures and different frame rates for the base and the enhancement layer. Frame 0 belongs to the previous GOP but is used in the prediction of the current GOP.

7.2.2 Temporal scalability

The enhancement layer pictures are typically coded as B-pictures, where the reference picture lists 0 and 1 are restricted to the temporally preceding and succeeding picture, respectively, with a temporal layer identifier less than the temporal layer identifier of the predicted picture. In the figure, frames 0 and 16 are coded first as intra-frames, then frame 8 is coded as a B-picture using frame 0 and 16 as references. Then the rest of the frames are coded in the order shown in the figure. As an example when frame 10 is coded, up to 3 references can be used in list 0 (left references), and 2 references in list 1 (right references). An additional bi-prediction signal, formed by a weighted sum of list 0 and list 1, can be used. In the JSVM software [42] the exact number of references is a user-configurable parameter.

7.2.3 Spatial scalability

In each spatial layer, MCP and intra-prediction are employed similar to the processes used for the base layer. Additionally an inter-layer prediction mechanism is incorporated as shown in figure 7.1. For instance, when frame 10 is coded, in addition to the 5 possible references described in the temporal scalability, information from the corresponding frame in the base layer can be used in MCP.

7.2.4 Quality scalability

Quality scalability is considered as a special case of spatial scalability with identical picture sizes for base and enhancement layers, therefore the same motion-compensated predictions including the inter-layer prediction are employed.

7.3 Related Work

To reduce the implementation complexity, several fast techniques have been presented recently [96, 100]. Most of them share the same concept of using the correlation between a MB and its neighbours in different layers, and also that between a MB in the base layer and its corresponding position in the enhancement layer.

In [96], a fast mode decision algorithm for inter-frame coding for temporal, spatial and quality scalability is presented. It makes use of the mode-distribution correlation between the base layer and enhancement layers. Specifically, after the exhaustive search technique is performed at the base layer, the number of candidate modes for enhancement layers is reduced.

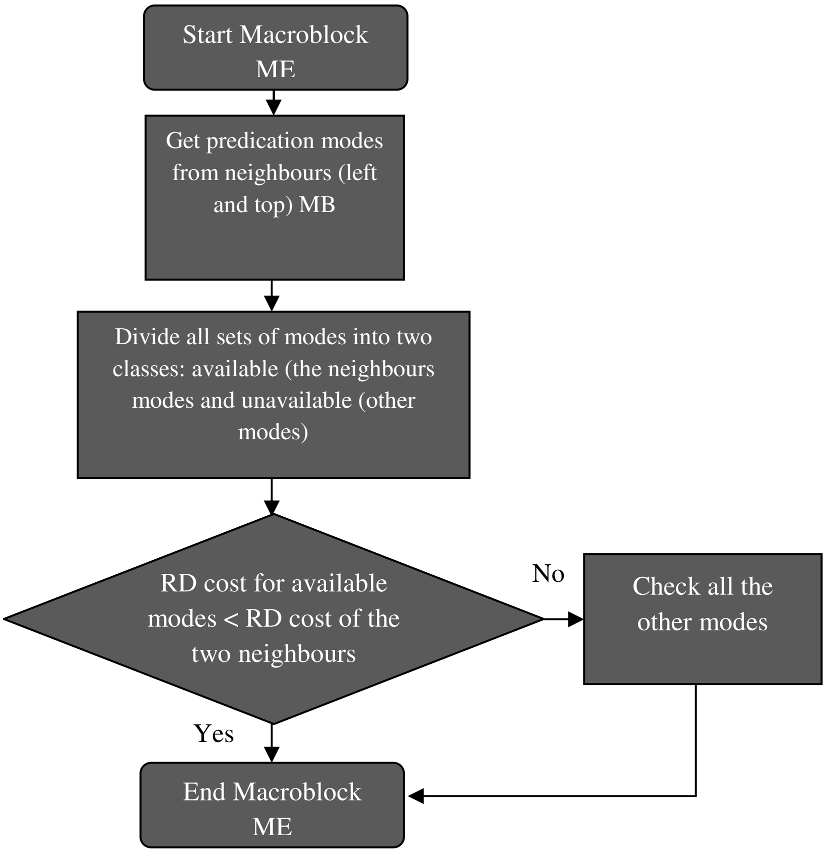

Start Macroblock ME
Get predication modes from neighbours (left and top) MB
Divide all sets of modes into two classes: available (the neighbours modes and unavailable (other modes)
RD cost for available modes < RD cost of the two neighbours
No
Check all the other modes
Yes
End Macroblock ME

Figure 7.2 – Ren's fast adaptive early termination MD for H.264/SVC [98].

In [97] and [98] fast mode decision algorithms exploiting correlations between MBs and their neighbours are proposed. These algorithms have an obvious limitation when applied to fast video sequences or sequences with complex backgrounds, as the irregularity between the MB and its neighbours increases, which results in limited benefits and significant performance degradation. The algorithm presented in [98] is shown in figure 7.2 as a flowchart.

In [99] a scheme that generates a reduced candidate mode list for the current MB according to the corresponding reference MBs was presented. It works by calculating the average PSNR of all frames at the same temporal levels for a specific mode, and then this PSNR is subtracted from the PSNR of the same mode for the current MB and compared to a threshold. Based on the result an early termination decision is made. A simplified implementation of this algorithm is shown in figure 7.3 as a flowchart.

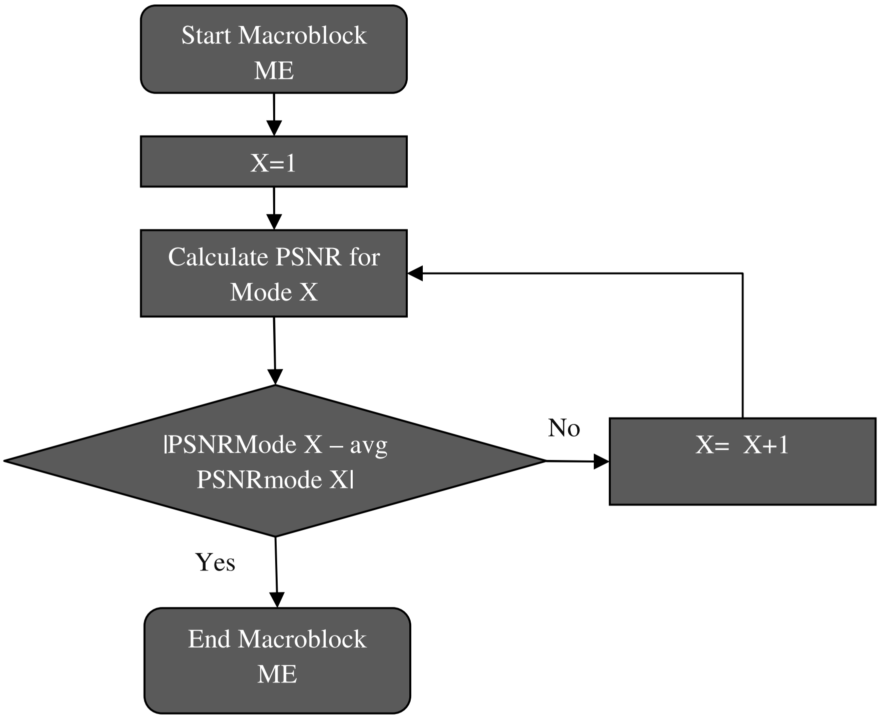


Figure 7.3 – Dewier's fast MD alogrthim for H.264/SVC [99].

This scheme shares the same limitation with [97] and [98] when applied to fast video sequences as the MB and the corresponding reference MBs are likely to have different modes.

In [100] it has been observed that there is a high correlation between the MB and its enclosed partitions when using forward (FW), backward (BW) and bi-directional (BI) predictions. Based on this correlation a three step algorithm has been proposed:

- First, an exhaustive search is carried out for the 16×16 MB including FW, BW and BI search.

- Secondly, based on the result from the first step the BI prediction is selectively eliminated for the 16×8, 8×16 and 8×8 partitions.

- Finally, a similar step to the second one is implemented to selectively reduce the prediction modes for the 8×4, 4×8 and the 4×4 partitions based on the 8×8 results. The algorithm is shown in figure 7.2 as a flowchart.

The algorithm can provide some time saving whilst maintaining a good RD performance, however, this time saving is limited to one prediction type (the BI prediction type) especially for the large block partitions. Furthermore, this prediction time is the least used type.

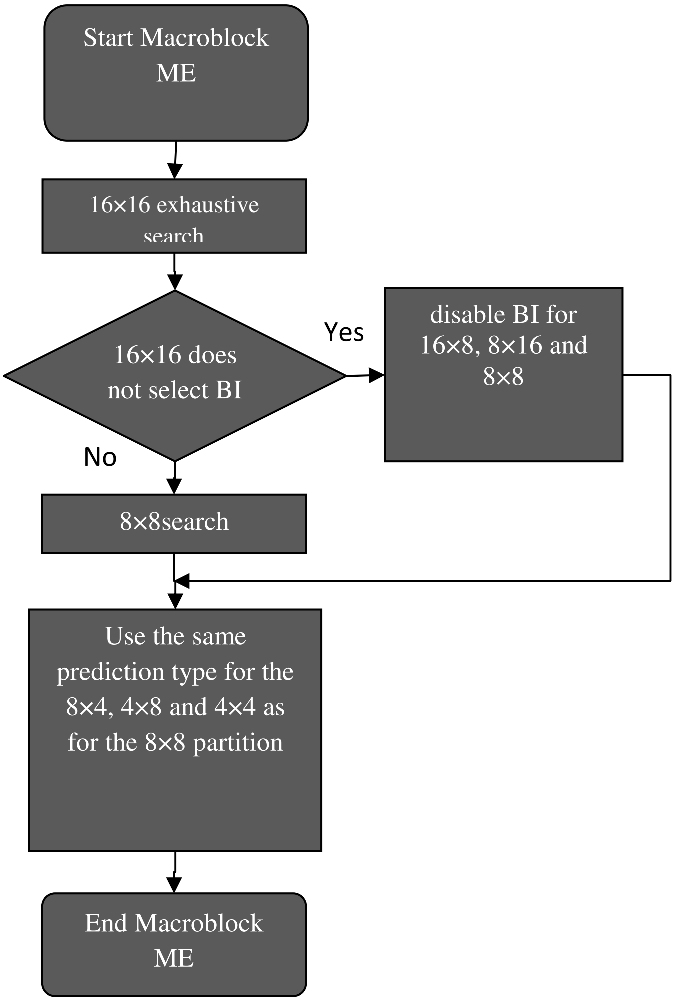

Start Macroblock ME
16×16 exhaustive search
16×16 does not select BI
Yes
disable BI for 16×8, 8×16 and 8×8
No
8×8search
Use the same prediction type for the 8×4, 4×8 and 4×4 as for the 8×8 partition
End Macroblock ME

Figure 7.4 – Lin's fast temporal prediction for H.264/SVC [100]

7.4 Proposed Algorithm

In the previous chapter the observation was made that there is a high correlation between the MB and its enclosed partitions when estimating the motion at different resolutions. Therefore a two step fast sub-pixel ME scheme based on this observation has been developed.

1. In the first step, if the 16×16 MB finds a best match in the full-pixel motion search that does not change after performing the sub-pixel motion search (condition one), then the sub-pixel motion search for all the enclosed 16×8 and 8×16 blocks is disabled.

2. Similarly, if in the second step the 8×8 block partitions of the 16×16 MB find the same best match in the full and sub-pixel motion searches (condition two), the sub-pixel motion search for all the enclosed 8×4, 4×8 and 4×4 blocks is disabled.

In the following sub-sections, several conditional probabilities that relate Condition one (C_1) and Condition two (C_2) to the ME and the mode distribution at different layers of the H264/SVC have been evaluated. Moreover, additional conditional probabilities of the mode distribution have been calculated.

7.4.1 Probability of mode selection in temporal scalability

The initial conditional probabilities focus on measuring the probability of the mode distribution at the different temporal layers in relation to C_1 and the mode distribution of lower temporal layers. Two probabilities were analysed:

P_1, the probability of the best mode belonging to the following set {Skip = Mode 0, 16×16 = Mode 1, 16×8 = Mode 2 and 8×16 = Mode 3) when C_1 is true.

P_2, the probability of the best mode belonging to the same set (less than 4) given that the best modes for all the corresponding MBs in list 0 and list 1 at lower temporal layer in the same GOP are less than 4.

This analysis was conducted using a GOP of size 16 for the different temporal layers. It reflects the average of using 9 Quantisation Parameters (QP) (10, 14, 18, 22, 26, 30, 34, 38, 42) as shown in table 7.1.

Sequence	QP	P_1 (%)	P_2 (%)			
			T1	T2	T3	T4
City	10, 14, 18, 22, 26, 30, 34, 38, 42	94.39	95.29	96.02	95.75	97.67
Harbour		97.02	94.06	98.97	99.13	91.73
Hall		99.28	99.53	99.68	98.54	99.57
Foreman		95.43	96.71	94.37	94.66	92.57
Mobile		96.13	92.16	95.68	97.24	92.33
Silent		99.67	99.88	99.98	98.65	99.78
News		98.79	98.69	98.02	96.87	97.76

Table 7.1 - Conditional probabilities of P_1 and P_2 expressed in terms of percentage

From the table it can be seen that the conditional probabilities are always higher than 90%. Therefore, P_1 and P_2 can be used as reliable indicators for mode prediction.

7.4.2 Correlation of mode selection in Spatial and SNR scalability

A second set of conditional probabilities was evaluated to calculate the probabilities of the mode distribution in the enhancement layers in relation to C_1 and the mode distribution in the base layer for spatial and SNR scalability. In the study three probabilities were analysed:

P_3, the probability of the best mode in the enhancement layer being less than 4 given that C_1 is true for the corresponding MB in the base layer.

P_4, the probability of the best mode in the enhancement layer being less than 4 given that the best mode of the corresponding MB in the base layer is less than 4.

P_5, the probability of the best mode in the enhancement layer being equal to the best mode of the corresponding MB in the base layer.

Table 7.2 shows an average of P_3, P_4 and P_5 when; GOP = 16, enhancement and base frame rate = 15, base QP = 14-42 and enhancement QP =12-40.

Sequence	Spatial			SNR		
	P_3 (%)	*P_4 (%)*	*P_5 (%)*	*P_3 (%)*	*P_4 (%)*	*P_5 (%)*
City	94.39	99.19	33.88	85.09	84.37	50.02
Harbour	97.02	98.82	26.81	80.14	80.55	42.25
Hall	99.98	99.28	13.53	98.23	99.8	19.19
Foreman	96.11	98.15	18.21	82.62	83.67	37.5
Mobile	95.76	97.23	23.62	81.19	81.22	22.13
Silent	99.69	99.93	36.18	95.46	98.35	51.68
News	99.99	99.65	33.36	96.78	98.71	53.17

Table 7.2 - Conditional probabilities of P_3, P_4 and P_5 expressed in terms of percentage

The table shows high correlation between the MB motion characteristics in the base layer and the best mode in the enhancement layers; this is reflected by the high P3 value. Also, the mode distribution characteristics of the enhancement layer are highly correlated to mode distribution characteristics of the base layer and should be used to predict the mode in the enhancement layer, this is shown by the high P_4 value. In [96] an observation related to P_4 has been used to limit the mode selection process. The low value of P_5 demonstrates that no assumptions can be made on the base layer and the enhancement layer having the same best mode.

7.4.3 Motion characteristics probabilities in spatial and SNR scalability

A final set of conditional probabilities were evaluated to calculate the probabilities of the motion characteristics in the enhancement layers in relation to C_1 and the C_2 in the base layer. Two probabilities were calculated:

P_6, the probability of the 16×16 in the enhancement layer C_1 being true given that C_1 for the corresponding MB in the base layer is true.

P_7, the probability of the 8×8 partition in the enhancement layer C_2 being true when C_2 for the corresponding block in the base layer is true.

Similar experimental conditions as in section 7.4.2 were used in the evaluation. The probabilities are shown in table 7.3 below.

Sequence	Spatial		SNR	
	P_6 (%)	P_7 (%)	P_6 (%)	P_7 (%)
City	87.60	89.62	85.09	83.09
Harbour	92.86	95.28	87.54	95.15
Hall	85.80	84.86	84.78	83.09
Foreman	82.15	86.26	83.31	81.72
Silent	97.77	93.57	89.14	93.26
News	95.23	97.48	91.38	97.66
Mobile	86.43	82.39	88.95	83.17

Table 7.3 - Conditional probabilities of P_6 and P_7 expressed in terms of percentage

The table shows that, on average 88% of the time when integer motion occurs in the base layer similar motion occurs in the enhancement layers. Thus, P_6 and P_7 can be used to adaptively enable and disable the sub-pixel motion search.

7.4.4 Low complexity hierarchical prediction algorithm

Based on the above observations, a fast mode decision algorithm and fast sub-pixel ME algorithm for inter-frame prediction was developed as follows:

A) Temporal scalability:

1. If the 16×16 find a best match in the full-pixel ME that does not change after performing the fractional pixel ME (integer-pixel motion); then:

    - limit the mode to 16×16, 16×8 and 8×16

    - disable the sub-pixel ME for the 16×8 and 8×16.

2. If the collective mode in the previously encoded MB in lower temporal layers is less than 4, then limit the mode selection to 16×16, 16×8 and 8×16.

3. If none of the above is true and if the 8×8 find a best match in the full-pixel ME that does not change after performing the fractional pixel ME (Integer-pixel Motion); then disable the sub-pixel ME for the 8×4 , 4×8 and 4×4

B) Spatial and SNR scalability:

In addition to 1), 2) and 3) in the temporal scalability the following three steps are added:

4. If the 16×16 MB in the base layer finds a best match in the full-pixel ME that does not change after performing the fractional-pixel ME (integer- pixel motion) then:

    - limit the mode to 16×16, 16×8 and 8×6,

    - disable the sub-pixel ME for the 16×8 and 8×16.

5. If the best mode for the corresponding block in the base layer is less than 4, then limit the mode selection to 16×16, 16×8 and 8×16.

6. If none of the above is true and if the corresponding 8×8 partition in the base layer find a best match in the full-pixel ME that does not change after performing the fractional-pixel ME (integer-pixel motion);then disable the sub-pixel ME for the 8×4 , 4×8 and 4×4.

The algorithm is summarised in figure 7.5 using pseudocode and in figure 7.6 using flowchart.

```
BEGIN
IF (MB ∈ Spatial or SNR Base Layer)
{
   IF (C1)
        {
           MODECURRENT_MB < 4;
  Disable sub-pixel Motion Search for the
  enclosed 16×8 and 8×16 blocks;
 }
      ELSIF (All the best modes in the previously encoded
             MBs in lower temporal layers < 4)
{
 MODECURRENT_MB   < 4;
 }
      ELSIF (C2)
 {
  Disable sub-pixel Motion Search for the
  enclosed 8×4, 4×8 and 4×4 blocks;
 }
}
ELSIF (C1 || C1 for corresponding Base Layer
           MB)
{
MODECURRENT_MB < 4;
Disable sub-pixel Motion Search for the enclosed 16×8   and 8×16 blocks;
}
ELSIF (the best mode of the corresponding Base Layer
         MB <  4)
{
 MODECURRENT_MB < 4;
}
ELSIF (C2|| C2 for corresponding Base Layer MB)
{
 Disable sub-pixel Motion Search for the enclosed 8×4, 4×8 and 4×4 blocks;
}
Decide best mode using the method described in section 2;
RETURN;
END
```

Figure 7.5 – Pseudocode of the proposed low complexity hierarchical prediction algorithm for H.264/SVC

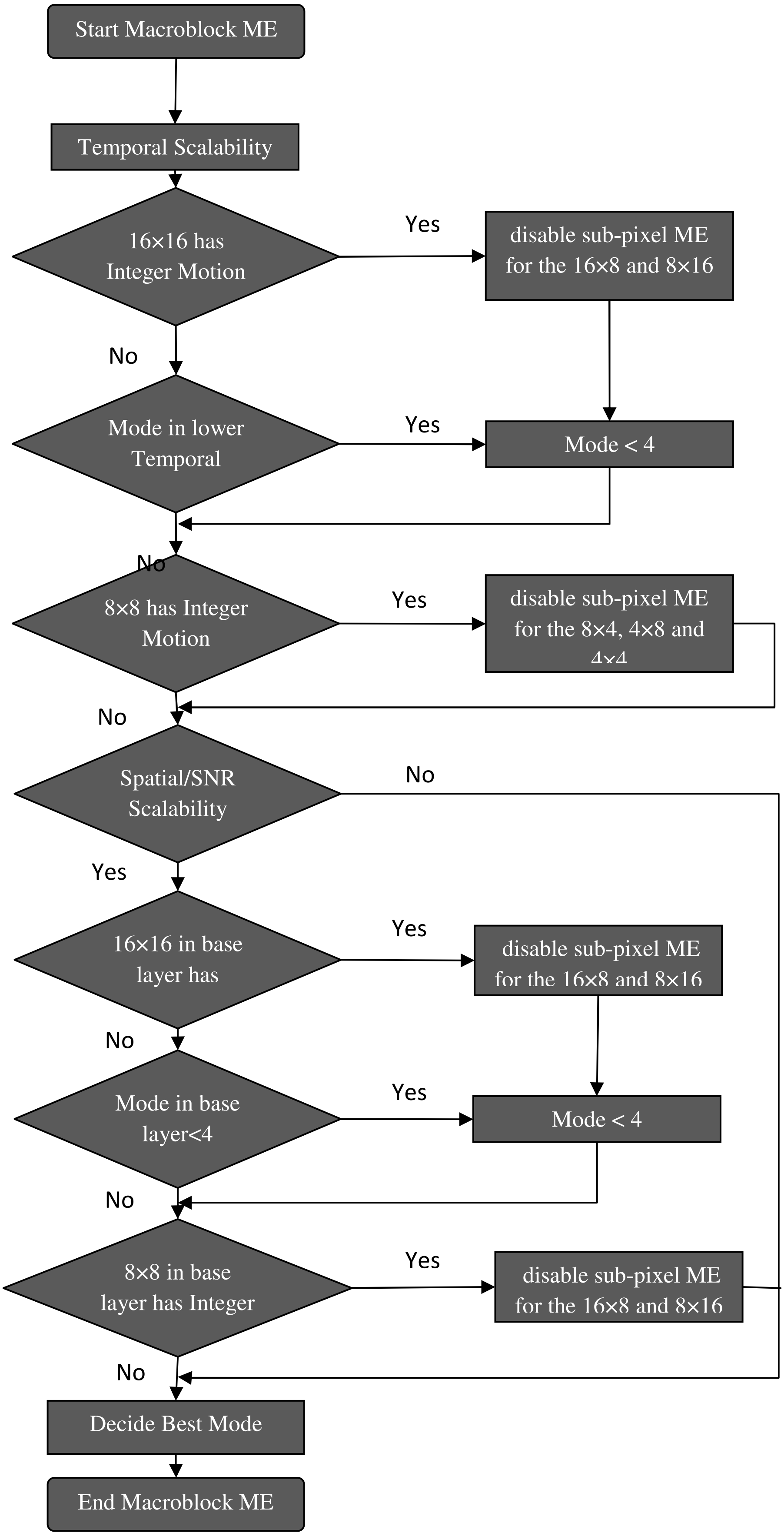


Figure 7.6 - The proposed low complexity hierarchical prediction algorithm for H.264/SVC

7.5 Experimental Results

To evaluate the proposed algorithm, a comprehensive set of experiments have been carried out. The experiment conditions are discussed in chapter 3.The encoder configuration is shown in Table 7.4. In the table this quantisation parameter range has been chosen to cover a wide bitrate range.

Parameter		Value		Parameter	Value
Resolution	Base	QCIF		**GOP size**	16
	Enhancement	CIF		**MV resolution**	¼ Pel
QP Setting	Base	14,22,30,38		**No. of frames**	100
	Enhancement	12,20,28,36		**Motion Search range**	16
Frame rate in/ out	Base	15HZ		**Reference Picture**	5
	Enhancement			**Search Function**	SAD

Table 7.4 - Encoder Experiment Conditions

7.5.1 Temporal Scalability

The experimental results are given in Table 7.5, in the table the negatives signs denote PSNR degradation and bitrate savings respectively. It can be seen that the proposed scheme achieves an average of 48.5% time saving with negligible losses in PSNR and increments in bitrate.

Sequence	Proposed T(s)	JSVM T(s)	TS (%)	BDPSNR (dB)	BDBR (%)
City	154.42	265.32	41.8	-0.09	1.89
Foreman	152.67	241.94	36.9	-0.15	2.5
Mobile	198.46	326.41	39.2	-0.16	1.5
Harbour	128.47	225.78	43.1	-0.12	1.1
Silent	65.21	149.24	56.3	-0.11	2.3
News	55.62	139.74	60.2	-0.1	2
Hall	53.98	143.2	62.5	-0.14	2.6
Average			48.5	-0.12	1.98

Table 7.5 - Comparison between the Proposed Algorithm and the JSVM 9.15 software.

Additionally the proposed algorithm has been compared with a recently proposed algorithm [100]. The comparison is shown in table 7.5, from the table it can be seen that for a very similar RD performance the proposed algorithm achieves an average of 27.8% time saving. It is important to notice that if the result in table 7.5 is compared to the result in [100], the result is not directly reflected in table 7.6. The reason for this is in the paper the authors used a high range of quantisation parameters (28, 32, 36 and 40) which makes the saving more significant. In contrast, in table 7.6 the QP range were (14, 22, 30 and 38), this range gives a better evaluation of both methods.

Sequence	TS (%)	BDPSNR (dB)	BDBR (%)
City	10.2	-0.01	0.05
Foreman	13.4	+0.02	0.01
Mobile	14.7	-0.01	-0.06
Harbour	17.2	+0.01	0.05
Silent	22.5	-0.21	+0.1
News	24.3	-0.1	+0.2
Hall	27.8	+0.14	0.01
Average	18.5	-0.02	0.05

Table 7.6 - Comparison between the proposed algorithm and Lin's algorithm [100]

7.5.2 Spatial and CGS scalability

In this experiment, the proposed algorithm is initially compared to the mode decision algorithm in JSVM9.15 software; the comparison result is shown in table 7.7. It can be observed from this table that our algorithm provides a time reduction in the range of 35–56% and 36–62%, depending on the video content, while generating the same video quality for scalable spatial and scalable quality video coding respectively.

Sequence	Spatial			SNR		
	TS (%)	BDPSNR(dB)	BDBR (%)	TS (%)	BDPSNR (dB)	BDBR(%)
City	37.2	-0.08	2.16	41.5	-0.09	1.77
Foreman	36	-0.1	2.4	36.5	-0.1	3.06
Mobile	35.4	-0.1	2	39.1	-0.	1.7
Harbour	37.7	-0.05	1.2	44.2	-0.08	1.5
Silent	48.9	-0.1	2.3	56.1	-0.14	2
News	51.5	-0.09	2	58.1	-0.12	1.8
Hall	56.6	-0.05	2.1	62.2	-0.07	2.2
Average	43.3	-0.08	2.02	48.2	-0.09	2

Table 7.7 - Comparison between the proposed algorithm and the JSVM 9.15 software

Sequence	Spatial			SNR		
	TS (%)	BDPSNR(dB)	BDBR (%)	TS (%)	BDPSNR(dB)	BDBR(%)
City	24.6	+0.25	-1.75	19.8	+0.08	-1.1
Foreman	7.75	+0.14	-2.27	4.4	+0.13	-1.7
Mobile	18.4	+0.07	-0.58	8.6	+0.08	-0.87
Harbour	28.7	+0.03	-0.2	26.4	+0.09	-2.32
Silent	10.7	+0.07	+0.11	13.1	+0.09	-1.3
News	9.5	+0.34	-2.09	10.2	+0.21	-2.07
Hall	30.1	+0.17	-0.6	11.7	+0.15	-0.69
Average	18.5	+0.16	-1.1	13.4	+0.11	-1.44

Table 7.8 - Comparison between the proposed algorithm and Ren's algorithm [98]

Table 7.8 provide provides comparative results for a recently developed fast mode selection algorithm [98] and the proposed method. It shows that our method achieves an average of 18.5% and 13.4% time saving for scalable spatial and scalable quality video coding respectively, while maintaining better PSNR and lower bitrate.

Sequence	TS(%)	BDPSNR(dB)	BDBR(%)
City	23.22	-0.02	1.1
Foreman	19.72	0.1	-1.7
Mobile	16.37	-0.08	0.87
Harbour	12.88	0.06	0.32
Silent	25.7	0.03	-1.4
News	29.4	0.01	0.07
Hall	32.8	-0.15	-0.69
Average	22.87	-0.007	0.2

Table 7.9 - Comparison between the proposed algorithm and Dewier's algorithm [99]

Table 7.9 provides comparative results for a recently developed fast mode selection algorithm [99] and the proposed method. It shows that proposed method achieves an average of 22.8% time saving while maintaining similar PSNR and bitrate.

7.6 Summary

In the SVC extension of the H.264/AVC standard, an exhaustive search technique is used to select the best coding mode for each MB. This technique achieves the highest possible coding efficiency, but it demands a very high computational complexity which constrains its use in many practical applications. This chapter proposes combined fast sub-pixel ME and a fast mode decision algorithm for inter-frame coding for temporal, spatial, and coarse grain signal-to-noise ratio scalability. It makes use of correlation between the MB and its enclosed partitions at different layers.

Experimental results show that the scheme reduces the computational complexity significantly with negligible coding loss and bitrate increases when compared to JSVM 9.15 and recently reported fast mode decision algorithms. This saved computation can advance the progress in the realisation of the H.264 scalable extension in real-time applications and low complexity coding systems.

This algorithm enables a novel computational complexity control of an H.264/SVC encoder. This algorithm is based on the complexity reduction algorithm described in chapter 6. This chapter has contributed to the body of knowledge by two publications [95] and [96].

CHAPTER 8
FAST MULTILAYERED PREDICTION ALGORITHM FOR H.264/MVC

8.1 Introduction

With the wide expansion of 3D and free viewpoint video applications, the H.264 Multiview Video Coding (MVC) standard has been developed as an extension to the H.264 AVC standard to enable efficient coding of scenes captured from multiple cameras [101]. Since all cameras capture the same scene from different viewpoints, inter-view statistical dependencies can be expected. Therefore, in addition to the H.264/AVC very refined ME and MC processes, H.264/MVC exploits inter-view prediction for more efficient coding. However, this further increases overall encoding complexity.

Recently proposed algorithms reduce encoder complexity by locating corresponding objects in neighbouring views by means of a global disparity vector and exploiting the mode distribution correlation between neighbouring views [102-107]. These algorithms perform well only for certain video sequences and camera configurations; given the inherent that scene characteristics are not taken into account.

In this chapter, the high correlation between a MB and its enclosed partitions is utilised to estimate motion homogeneity, and based on this result inter-view prediction is selectively enabled or disabled. Moreover, the MVC is divided into three layers in terms of motion prediction; the first being the full and sub-pixel motion search, the second being the mode selection process and the third being repetition of the first and second for inter-view prediction, the proposed algorithm significantly reduces the complexity in the three layers. This is accomplished by extending the algorithm proposed in the previous chapter [94] and applying it to the inter-view prediction [108].

This chapter is organised as follows: The MVC concepts and requirements are outlined in section 8.2. Section 8.3 reviews some related work on reducing computational complexity in MVC prediction process. The proposed algorithm is presented in Section 8.4. Experimental results are presented in Section 8.5. Finally, a summary is provided in Section 8.6.

8.2 H.264/ Multiview Video Coding (MVC)

8.2.1 Overview

Multiview Video Coding (MVC) is an extension to the H.264/MPEG-4 AVC video compression standard developed with joint efforts by MPEG/VCEG to enable efficient encoding of sequences captured simultaneously from multiple cameras using a single video stream. Therefore the design is aimed at exploiting inter-view dependencies in addition to reducing temporal redundancies [101, 109].

The stereo high profile of the MVC standard has been standardised in June 2009. MVC streams are backward compatible with H.264/AVC, which allows for older devices and software to decode stereoscopic video streams, ignoring additional information for the second view [15].

8.2.2 MVC Targeted Applications

The MVC main targeted applications are:

1. Stereoscopic: two-view video

2. Free Viewpoint Television (FTV): a system allowing the user to control the viewpoint and add new views of a scene from any 3D position.

3. Multi-view 3D television: a television that designed to display 3D materials. Those materials can be formed using any 3D production techniques , such as stereoscopic capture, multi-view capture, or 2D plus depth.

Two main challenges face most multiview applications; the first one is the transmission of huge amount of data, which requires the development of highly efficient coding schemes, and the second one is that any compression scheme designed specifically for multiview video streams should support random access functionality, allowing viewers to access arbitrary views with minimum time delay. Therefore, a set of requirements has been laid out for designing the MVC as explained in the following section.

8.2.3 Requirements

Most of the requirements as well as test data and evaluation conditions for the Multiview coding standard are defined by the MVC project [111], a summary of those is expressed in the following points.

1. Large gain compared to independent compression of each view.

2. Temporal random access and view random access.

3. View scalability, this means any part of the bitstream can be accessed by the decoder to generate a low-quality video output

4. Parallel processing, to reduce delays, implementation allows for the encoding of multiple views simultaneously.

5. Camera parameters (extrinsic and intrinsic) were required to be transmitted with the bitstream to support the main view interpolation.

6. Backward compatibility with the AVC.

7. Consistent quality amongst views.

8.2.4 Temporal and Inter-View Correlation

Several analyses [111, 112] have been carried out to investigate temporal and inter-view correlation by measuring the statistical dependencies that can be exploited for prediction.

Figure 8.1 shows the eight possible first order spatial and temporal neighbour pictures of a picture in a MultiView Video (MVV) sequence with linear camera arrangement, where V indicates the views and T the time-points (the temporal position).

If $F_{0,0}$ is considered to be the current frame, and the linear camera arrangement shown in figure 8.1 is considered; there are 9 possible frames that could be used as reference frames for motion prediction purposes as shown in the figure. $F_{0,-1}$ and $F_{0,1}$, represent the preceding and the succeeding frames in the same view respectively, and are normally considered as the reference frames in the H.264/AVC.

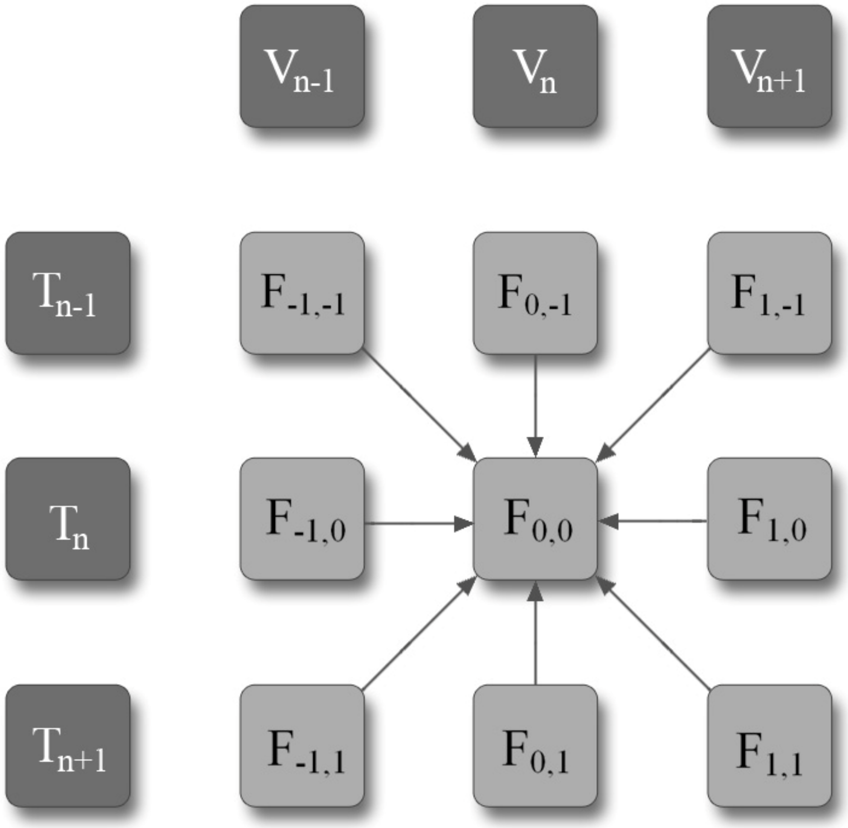


Figure 8.1 - Eight possible first order spatial and temporal neighbour pictures

In [111] and [112] the analysis results demonstrate that frequently for a significant number of MBs inter-view prediction is more efficient than temporal prediction, although for all video sequences temporal prediction is the most often chosen mode. This is due to illumination difference or imperfect calibration of cameras [113]. Inter-view prediction means that a MB in $F_{0,0}$ finds a best match in V_{n-1} or V_{n+1} , whilst temporal prediction means that MB in $F_{0,0}$ finds a best match in V_n.

A comparison between inter-view prediction and temporal prediction modes is shown in table 8.1. The comparison reflects the percentage between the MBs that find their best match in the same views or in neighbours' views.

Sequence Name	T [%]	V [%]
Ballroom	83.24	16.76
Exit	86.42	13.58
Uli	95.65	4.35
Race1	98.26	1.74
Breakdancers	70.93	29.07
Average	86.9	13.1

Table 8.1 - Results of temporal and inter-view correlation analysis

In table 8.1 standard multiview testing sequences were used (details of those sequences and experiments settings can be found in chapter 3, experimental method).

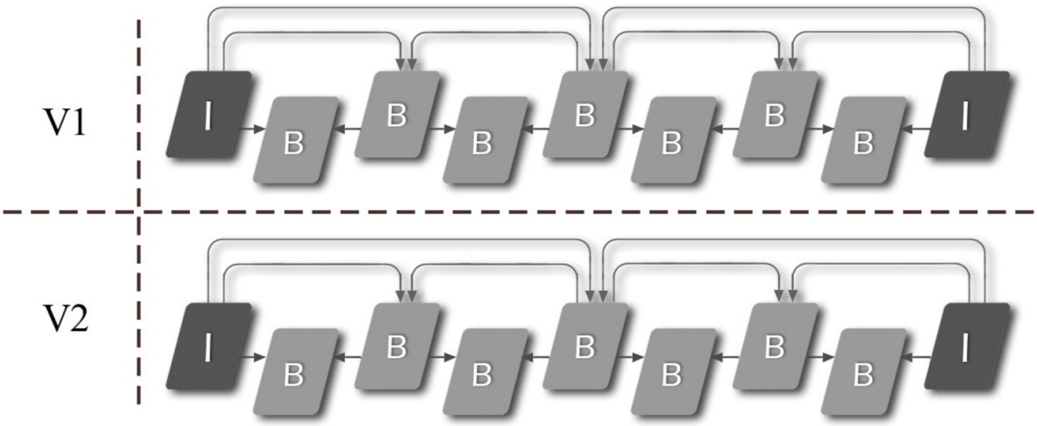
The table indicates that on average 13.1% of the MBs of all sequences find a best match in other views. This can lead to a considerable bitrate reduction but at the expense of increasing the encoder complexity. The following section outlines some of the proposed prediction structures for MVC.

8.2.5 Prediction Structures

Since MVC is a direct extension of AVC with the addition of the inter-view prediction, the MVC prediction structure is based on the multiple reference picture technique in H.264/AVC. Therefore in the design stage for the standard different prediction structures have been proposed [114-120]. Those structures vary significantly in terms of the overall performance and the encoder requirements for reference picture selection and memory management. The following subsections outline three of those variations that are enabled in the standard reference software.

8.2.5.1 Temporal Prediction using hierarchical B pictures

The simplest way to encode a set of video streams from different cameras is to encode them separately using the H.264/ AVC as shown in figure 8.2. The figure shows two GOPs for two views, each contains 8 frames. In the figure, the hierarchical B pictures method [121] is employed as it is considered the most efficient temporal prediction structure.



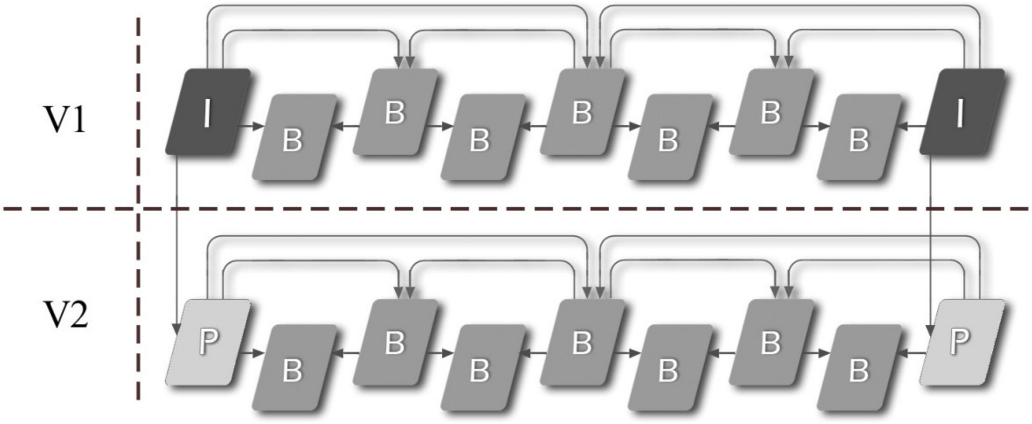
Display Order	1	2	3	4	5	6	7	8	9
Coding Order	0	5	3	6	2	7	4	8	1

Figure 8.2 - Temporal prediction using hierarchical B pictures

The first picture of a video sequence is intra-coded as IDR picture and so-called key pictures, referred to as I picture in figure 8.2. Then at regular intervals, defined by the GOP size, frames are coded as I frames. As discussed this method is simple but inefficient for multiview videos.

8.2.5.2 Inter-view Prediction for Key Pictures

A straight forward improvement to temporal prediction using hierarchical B pictures is to employ inter-view prediction for key pictures only as shown in figure 8.3.



Display Order	1	2	3	4	5	6	7	8	9
Coding Order	0	5	3	6	2	7	4	8	1

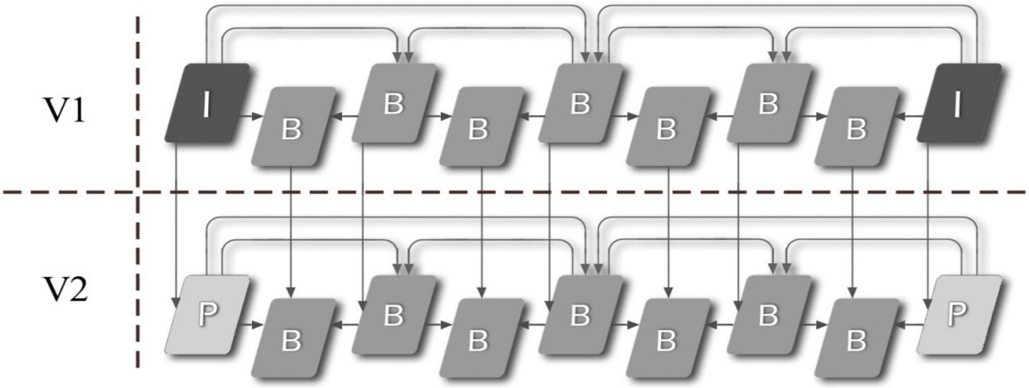
Figure 8.3 - Inter-view prediction for key pictures.

In this approach all I frames except those in the first view are encoded as P pictures. This can lead to a significant number of bit savings due to the fact that I frames require a lot more bits than P frames [121].

The downside of this method is that individual views can no longer be encoded or decoded independently as they share reference pictures. Furthermore, the design of the encoder and the decoder becomes more complex, specifically for managing reference frames and data buffers.

8.2.5.3 Inter-view prediction for key and non-key pictures

Another approach commonly used for MVC is the inter-view prediction for key and non-key pictures. This method is shown in figure 8.4.

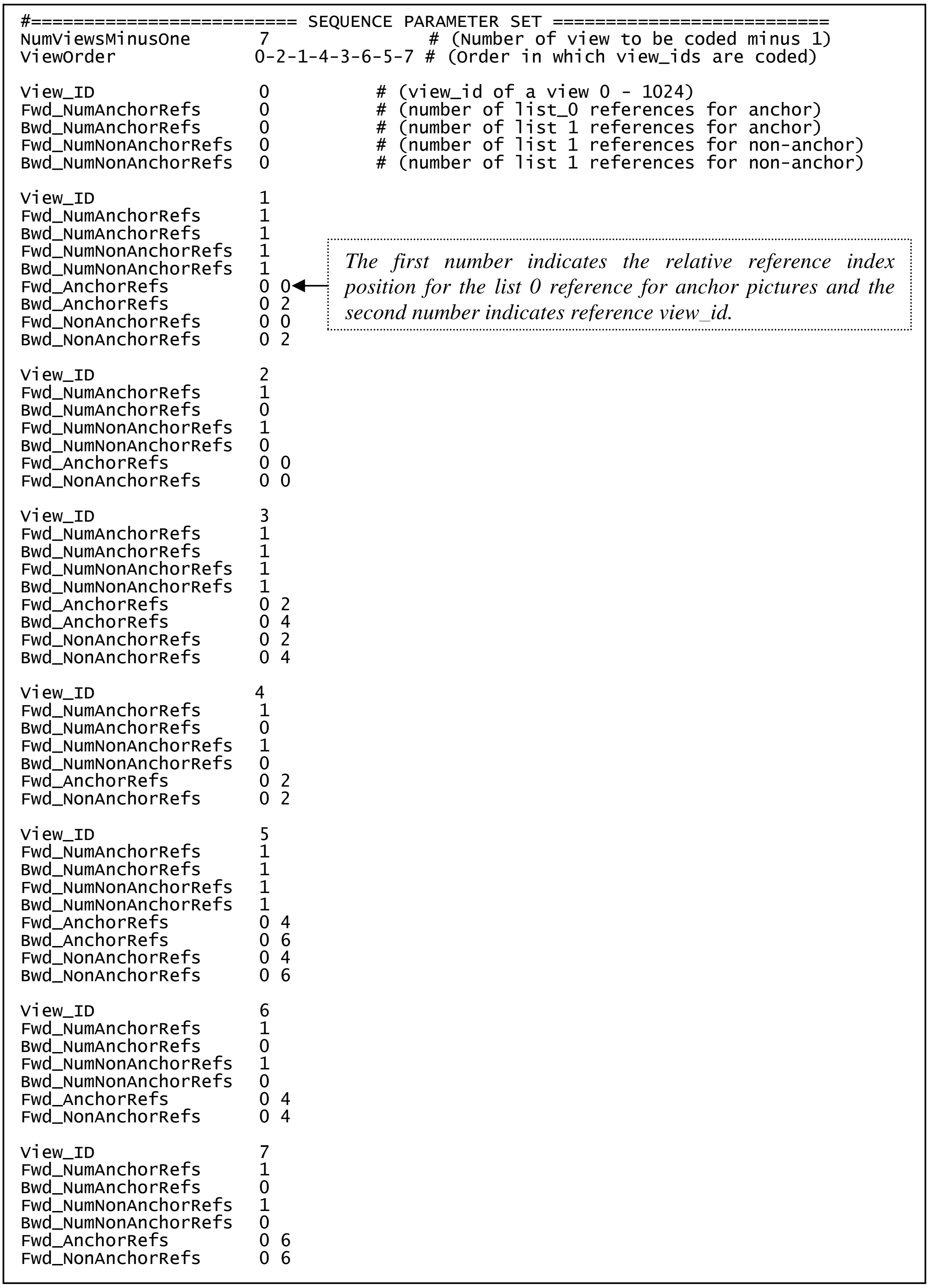


Display Order	1	2	3	4	5	6	7	8	9
Coding Order	0	5	3	6	2	7	4	8	1

Figure 8.4 - Inter-view prediction for key pictures and non-key pictures.

From the figure it can be seen that this method allows greater flexibly for motion prediction, however this is at the expense of increasing the overall encoding complexity. Using the method in figure 8.4 a coding gain of 1.7dB can be achieved [112].

The MVC standard reference software [43] allows great flexibility in encoding multiviews videos. JMVM uses hierarchical B pictures for each view, and, at the same time, applies inter-view prediction to every 2nd view, previously encoded frames from adjacent camera views. This is accomplished by employing a number of user-configurable parameters in the software main configuration file. An example of this configuration file is shown in figure 8.5.



```
#======================== SEQUENCE PARAMETER SET ========================
NumViewsMinusOne       7              # (Number of view to be coded minus 1)
ViewOrder              0-2-1-4-3-6-5-7 # (Order in which view_ids are coded)

View_ID                0         # (view_id of a view 0 - 1024)
Fwd_NumAnchorRefs      0         # (number of list_0 references for anchor)
Bwd_NumAnchorRefs      0         # (number of list 1 references for anchor)
Fwd_NumNonAnchorRefs   0         # (number of list 1 references for non-anchor)
Bwd_NumNonAnchorRefs   0         # (number of list 1 references for non-anchor)

View_ID                1
Fwd_NumAnchorRefs      1
Bwd_NumAnchorRefs      1
Fwd_NumNonAnchorRefs   1
Bwd_NumNonAnchorRefs   1
Fwd_AnchorRefs         0 0
Bwd_AnchorRefs         0 2
Fwd_NonAnchorRefs      0 0
Bwd_NonAnchorRefs      0 2

View_ID                2
Fwd_NumAnchorRefs      1
Bwd_NumAnchorRefs      0
Fwd_NumNonAnchorRefs   1
Bwd_NumNonAnchorRefs   0
Fwd_AnchorRefs         0 0
Fwd_NonAnchorRefs      0 0

View_ID                3
Fwd_NumAnchorRefs      1
Bwd_NumAnchorRefs      1
Fwd_NumNonAnchorRefs   1
Bwd_NumNonAnchorRefs   1
Fwd_AnchorRefs         0 2
Bwd_AnchorRefs         0 4
Fwd_NonAnchorRefs      0 2
Bwd_NonAnchorRefs      0 4

View_ID               4
Fwd_NumAnchorRefs      1
Bwd_NumAnchorRefs      0
Fwd_NumNonAnchorRefs   1
Bwd_NumNonAnchorRefs   0
Fwd_AnchorRefs         0 2
Fwd_NonAnchorRefs      0 2

View_ID                5
Fwd_NumAnchorRefs      1
Bwd_NumAnchorRefs      1
Fwd_NumNonAnchorRefs   1
Bwd_NumNonAnchorRefs   1
Fwd_AnchorRefs         0 4
Bwd_AnchorRefs         0 6
Fwd_NonAnchorRefs      0 4
Bwd_NonAnchorRefs      0 6

View_ID                6
Fwd_NumAnchorRefs      1
Bwd_NumAnchorRefs      0
Fwd_NumNonAnchorRefs   1
Bwd_NumNonAnchorRefs   0
Fwd_AnchorRefs         0 4
Fwd_NonAnchorRefs      0 4

View_ID                7
Fwd_NumAnchorRefs      1
Bwd_NumAnchorRefs      0
Fwd_NumNonAnchorRefs   1
Bwd_NumNonAnchorRefs   0
Fwd_AnchorRefs         0 6
Fwd_NonAnchorRefs      0 6
```

The first number indicates the relative reference index position for the list 0 reference for anchor pictures and the second number indicates reference view_id.

Figure 8.5 - Example of JMVM configuration for a 5 view video

In this example an 8-view video is encoded, from the figure, the user firstly inputs the number of views and the coding order. Then as shown in the figure the user has the flexibility to select which view is used as reference and how many references are used for key and non-key frames, given that the reference view is previously encoded. For example in this configuration, key pictures and non-key pictures in the third view will use 2 inter-view references from view 2 and 4.

8.3 Efficient Multiview Prediction Algorithms

Most of the fast prediction algorithms that were designed for and applied to H.264/AVC can be implemented in any view of the MVC views (some of those algorithms were discussed in chapter 4). However, as discussed in the previous section, due to the inter-view flexibility in the MVC, the number of the possible references for any frame is far more than the AVC.

Recently, a few algorithms [102-107] have been proposed to speed up the prediction process in the view direction. For example, in [104] a fast inter-frame prediction algorithm has been presented. It works by deciding whether or not the inter-view prediction is used for a MB based on its neighbours. For example, if the top and left co-located MBs find a best match using inter-view prediction, the MB to be coded use reference frames in the view directions, otherwise the MB uses reference frames in the temporal direction. Furthermore, if the RD cost is less than a particular threshold, references from the other direction are used. Additionally, three more steps were added to speed up the motion prediction and the mode decision process by making use of the camera parameters and their effect on the object position in different views. Although the algorithm works well for a number of video sequences such as Ballroom and Exit, the exploitation of thresholds to adaptively control the inter-view prediction has led to a significant bitrate increase for some sequences.

In [105], an algorithm that makes use of the mode distribution correlation between neighbour views has been proposed to enhance the complexity efficiency. A mode complexity parameter for the current MB is defined from the previously encoded co-located MB in neighbour views. Then, that parameter is compared to a threshold and based on the comparison results the MB can be categorised into one of three categories:

- MB with simple mode : all modes are terminated apart from the 16×16 mode.

- MB with medium mode : only the 16×16, 8×16 and 16×8 modes are examined.

- MB with complex mode : all modes are tested.

The algorithm is shown figure 8.6 as a flowchart.

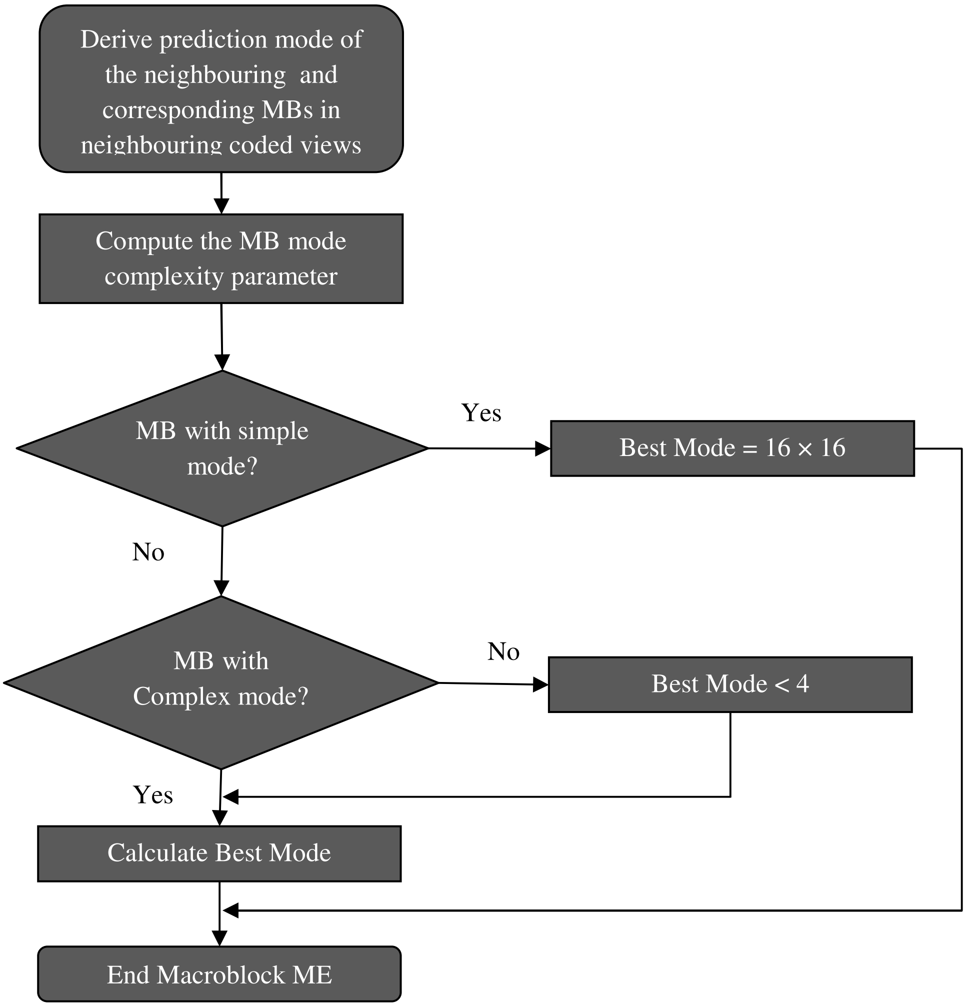

Derive prediction mode of the neighbouring and corresponding MBs in neighbouring coded views
Compute the MB mode complexity parameter
MB with simple mode?
Yes
Best Mode = 16 × 16
No
MB with Complex mode?
No
Best Mode < 4
Yes
Calculate Best Mode
End Macroblock ME

Figure 8.6 – Shen's fast MD for algorithm H.264/MVC [105]

This scheme is based on statistical analysis that proved the correlation between the co-located MBs in neighbour views. However, introducing thresholds to control the RD performance when compared to the standard reference software has led to limiting the gain in some sequences.

A similar algorithm to [105] has been proposed in [106], the basic idea of this method is to utilise the spatial property of the motion field to decide on when to use the inter-view prediction. Similar to [105] neighbouring MBs are used to predict the motion of the current MB. In the first step motion homogeneity around the MB is determined by comparing the average MV of the neighbour MBs and the corresponding one in different views to a threshold. If a MB is found to be within a homogeneous motion region, only the inter_16×16 and intra_16×16 modes are tested, otherwise all the modes are examined. The algorithm is shown in figure 8.7 as a flowchart.

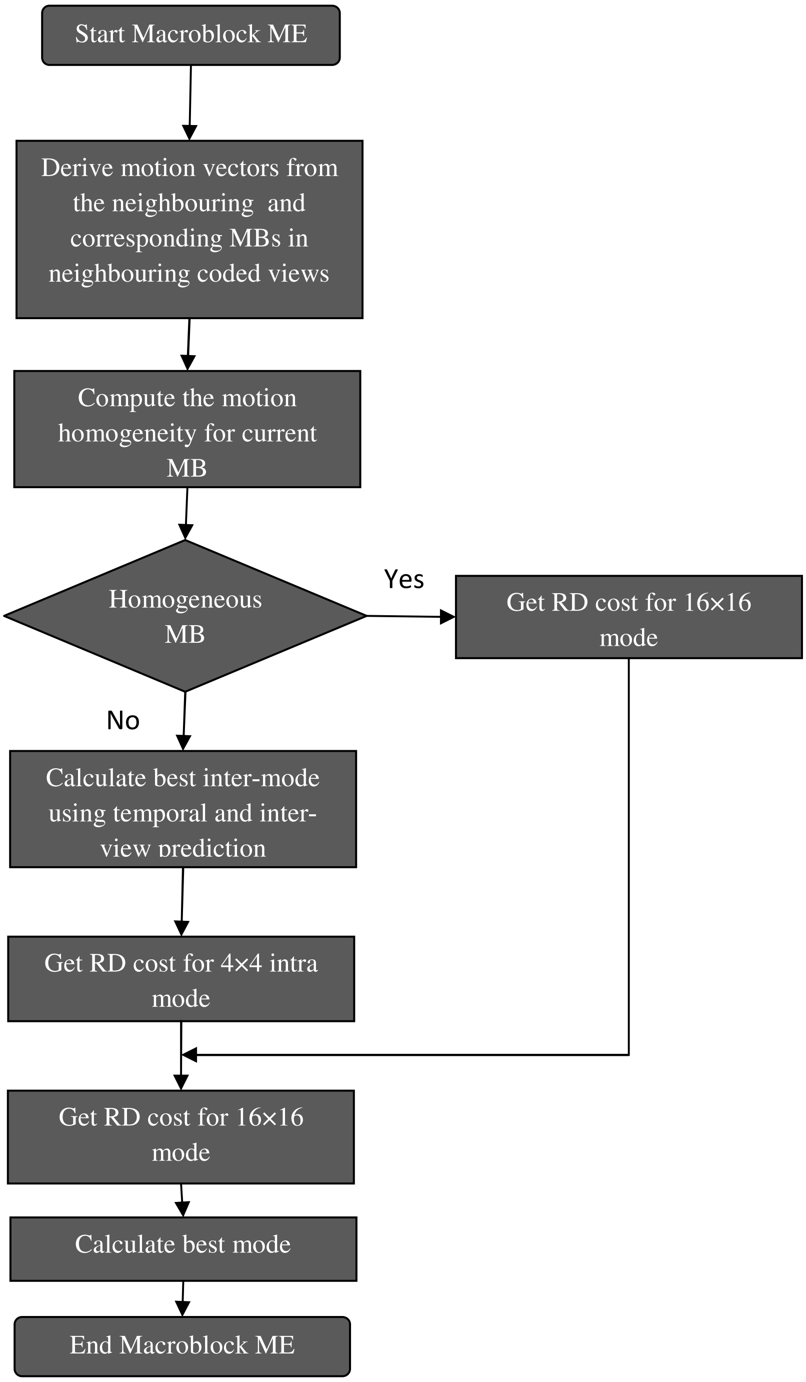

Start Macroblock ME
Derive motion vectors from the neighbouring and corresponding MBs in neighbouring coded views
Compute the motion homogeneity for current MB
Homogeneous MB
Yes
Get RD cost for 16×16 mode
No
Calculate best inter-mode using temporal and inter-view prediction
Get RD cost for 4×4 intra mode
Get RD cost for 16×16 mode
Calculate best mode
End Macroblock ME

Figure 8.7 – Shen's selective disparity estimation for H.264/MVC [106]

In [107], an object-based fast prediction mode decision method has been suggested. Segmentation is used to divide the frames into foreground and background objects. First, motion-based segmentation is applied to non-anchor frames by using information from both motion vectors and intensity value. Then, the disparity-based segmentation is carried out by considering the distribution of disparity vectors in the reference anchor picture. After the segmentation, inter-view prediction is only employed for MBs in the foreground regions.

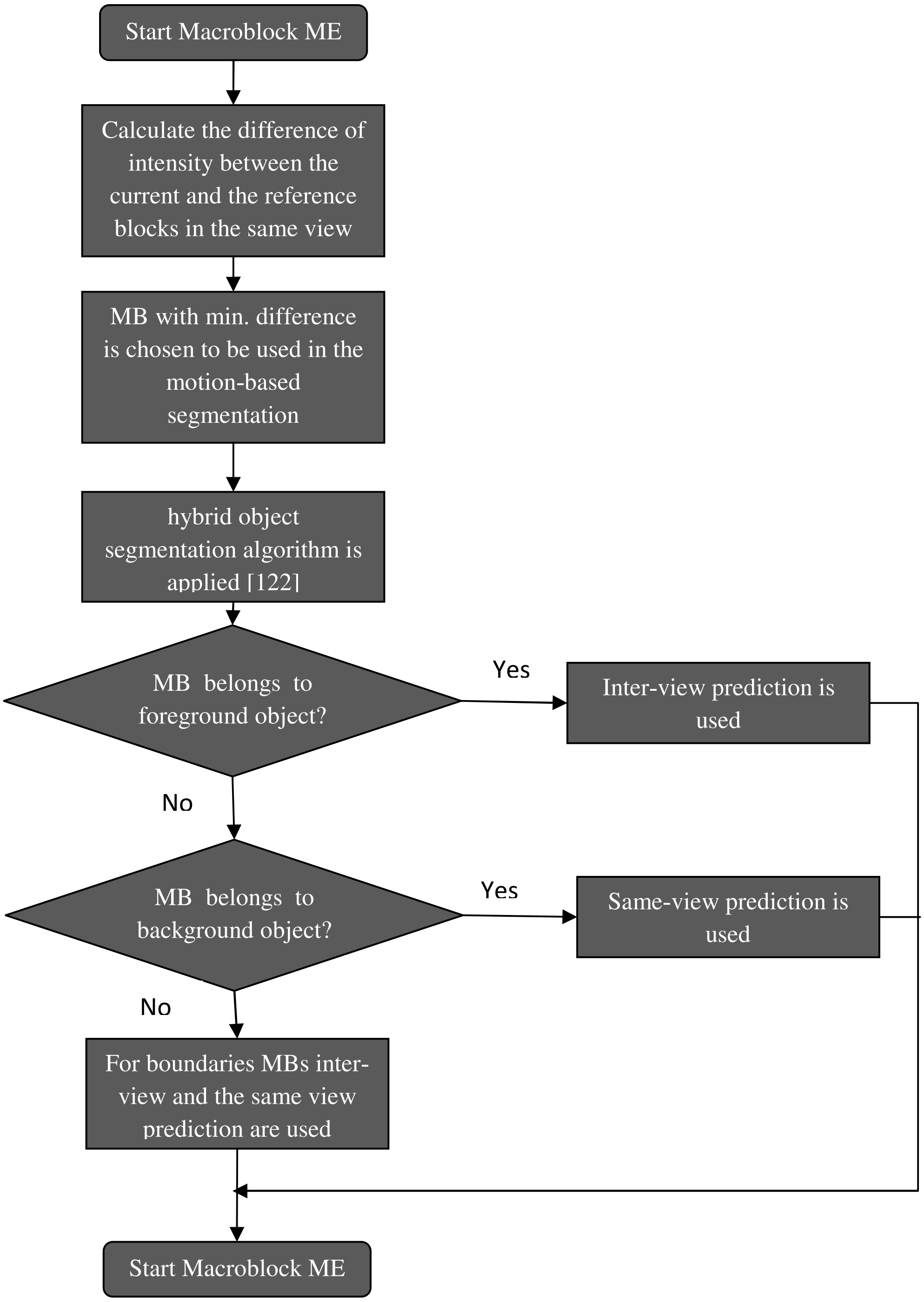

Start Macroblock ME
Calculate the difference of intensity between the current and the reference blocks in the same view
MB with min. difference is chosen to be used in the motion-based segmentation
hybrid object segmentation algorithm is applied [122]
MB belongs to foreground object?
Yes
Inter-view prediction is used
No
MB belongs to background object?
Yes
Same-view prediction is used
No
For boundaries MBs inter-view and the same view prediction are used
Start Macroblock ME

Figure 8.8 – Lee’s fast MD algorithm for H.264/MVC [107]

The algorithm is shown in figure 8.8 as a flowchart. From the figure it can be seen that the algorithm applies several complex pre-processing steps for the segmentation purpose which limits the overall gain.

8.4 Proposed Algorithm

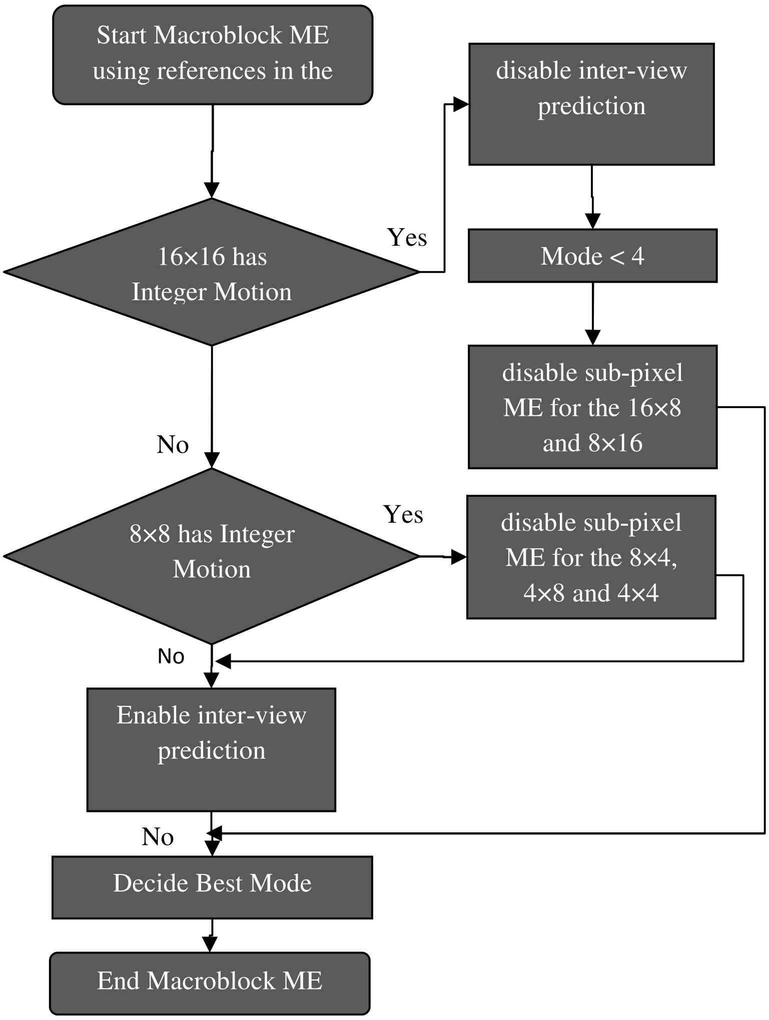
In this section a novel fast prediction algorithm is proposed [108]. In contrast to the different algorithms that were discussed in the previous section, the proposed algorithm does not depend on motion prediction results of the co-located MB in the same view or in different views. Furthermore, no thresholds are used to keep the balance between the RD cost and reducing the complexity. Instead the MB internal information is exploited to reduce the complexity.

If the motion prediction in the MVC is considered to be a three layers process; the first being the full and sub-pixel motion search, the second being the mode selection process and the third being repetition of the first and second for inter-view prediction, the proposed algorithm significantly reduces the complexity in the three layers. The first and second layers are the same for the AVC, SVC and MVC. The algorithm takes advantage of the proven fact [106-107] that only fast moving objects in any view tend to find their best matches in neighbour views. It also takes advantage of the fast motion and mode selection algorithm that has been proposed in chapter 7 [95]. Which utilises the correlation between a MB and it enclosed partitions' motion results in different layers to define areas in any frame with integer-motion. Those areas can also be classed as homogeneous areas. An additional step has been added to limit the use of inter-view prediction to fast moving objects, thereby reducing the overall complexity. The algorithm can be summarised in the following steps:

1. If the 16×16 finds a best match in the full-pixel ME that does not change after performing the fractional-pixel ME (integer-pixel motion);then i) Disable inter-view prediction, ii) limit the mode to 16×16, 16×8 and 8×16 and iii) Disable the sub-pixel ME for the 16×8 and 8×16.

2. If the 8×8 finds a best match in the full-pixel ME that does not change after performing the fractional-pixel ME (integer-pixel motion); then disable the sub-pixel ME for the 8×4, 4×8 and 4×4.

The algorithm is shown in figure 8.9 as a flowchart.



Start Macroblock ME using references in the
16×16 has Integer Motion
Yes
disable inter-view prediction
Mode < 4
disable sub-pixel ME for the 16×8 and 8×16
No
8×8 has Integer Motion
Yes
disable sub-pixel ME for the 8×4, 4×8 and 4×4
No
Enable inter-view prediction
No
Decide Best Mode
End Macroblock ME

Figure 8.9 - The proposed algorithm

The main advantage of this method in comparison to other schemes is that it makes use of some of the standard available tools to find homogeneity instead of employing additional pre-processing steps to segment the frames. Also, no additional statistical analyses need to be carried out and therefore statistical results do not need to be stored.

8.5 Experimental results

To evaluate the proposed algorithm, a comprehensive set of experiments have been carried out. The experiments conditions and setting are discussed in chapter 3.The encoder configurations are shown in Table 8.2.

Parameter	Value		Parameter	Value
Resolution	640×480		GOP size	16
	1024×768		MV resolution	¼ Pel
QP Setting	14-22-30-38		No. of frames	200-300
			Motion Search range	32
Frame rate in/ out	15-25 and 30 HZ		Reference Picture	2
			Search Function	SAD

Table 8.2- Experiment encoder configurations

Initially the algorithm was compared to the MVC reference software [43]. The experimental results are shown in table 8.3.

Sequence	Proposed T (s)	JMVM T (s)	TS (%)	BDPSNR (dB)	BDBR (%)
Akko&kayo	9717.93	21789.09	55.4	-0.09	1.09
Flamenco	6224.63	13415.15	53.6	-0.06	1.62
Race	13871.66	24251.15	42.8	-0.03	0.86
Rena	13054.65	38969.09	66.5	-0.1	1.04
Uli	6192.93	12983.08	52.3	-0.01	0.73
Ballet	3396.46	14223.03	76.12	-0.02	0.92
Breakdancing	1964755	53902.74	63.55	-0.1	0.93
Exit	10504.53	29842.42	64.8	-0.05	1.34
Average			59.41	-0.05	1.06

Table 8.3- Comparison between the proposed algorithm and the JMVM 8.0 software

It can be seen that the proposed scheme achieves an average of 59.38% time saving with negligible losses in PSNR and negligible increase in bit rate.

Additionally the proposed algorithm has been compared with the recently proposed algorithm [105-107]. The comparison is shown in table 8.4, 8.5 and 8.6 respectively.

Sequence	TS (%)	BDPSNR (dB)	BDBR (%)
Akko&kayo	6.2	0.05	-0.6
Flamenco	8.3	0.07	-1.10
Race	3.1	0.04	0.54
Rena	4.1	0.09	-1.23
Uli	16.4	0.05	0.085
Ballet	12.5	0.12	0.225
Breakdancing	11.7	0.08	-1.4
Exit	4.1	0.06	0.2
Average	8.3	0.07	-0.41

Table 8.4- Comparison between the proposed algorithm and Shen’s algorithm [105]

Sequence	TS (%)	BDPSNR (dB)	BDBR (%)
Akko&kayo	9.3	0.04	0.53
Flamenco	6.8	0.05	-1.03
Race	8.7	0.06	-0.7
Rena	14.6	-0.02	0.75
Uli	15.7	-0.03	0.55
Ballet	6.8	0.01	-0.09
Breakdancing	11.6	0.09	-0.6
Exit	9.5	-0.01	1.2
Average	10.38	0.023	0.076

Table 8.5- Comparison between the proposed algorithm and Shen's algorithm [106]

Sequence	TS (%)	BDPSNR (dB)	BDBR (%)
Akko&kayo	29.4	-0.01	0.15
Flamenco	27.8	-0.04	0.21
Race	19.6	0.06	0.27
Rena	25.6	-0.03	0.12
Uli	24.2	-0.02	1.01
Ballet	20.2	-0.01	-0.01
Breakdancing	18.5	0.05	-0.73
Exit	21.4	-0.11	-0.143
Average	23.33	-0.014	0.10

Table 8.6- Comparison between the proposed algorithm and Lee's algorithm [107]

From the tables it can be seen that for a very similar RD performance the proposed algorithm achieves an average of 8.3%, 10.38% and 23.33% time saving when compared to [105], [106] and [107] respectively. Although the time saving is video content dependent, the proposed scheme results in significant time savings when compared to H.264/MVC reference software and other known work.

8.6 Summary

The large amount of video data and particularly high computational complexity makes the MVC encoder difficult to be implemented in real-time applications. This chapter has presented a fast algorithm for multiview video coding based on the work in chapter 6 and chapter 7. In addition to reducing the complexity of the sub-pixel ME and the mode decision for frames in the same view, a novel early reference termination has been incorporated to the inter-view prediction of MBs. The proposed algorithm depends on the property of video sequences that fast moving objects are likely to be predicted using inter-view references while background objects are more likely to be predicted using references from the same view. The MBs inherited correlation is exploited as a motion-based segmentation to locate fast moving objects.

This is novel method gives the algorithm automated adaption to any video sequence with any characteristic, while most proposed fast algorithm in the area relay heavily on the spatial and the temporal correlation between MBs which limits their time saving to certain sequences. Thus the advantage of the resulting multiview prediction structure is achieving significant coding gains and being highly flexible regarding its adaptation to all kinds of spatial and temporal setups. This chapter has contributed to the body of knowledge by one publication [108].

To assess the proposed algorithm, a comprehensive set of experiments were conducted. The results show that the proposed algorithm significantly reduces the motion estimation time whilst maintaining similar rate distortion performance, when compared to both the H.264/MVC reference software and recently reported work.

CHAPTER 9
CONCLUSION

9.1 Introduction

Today, video coding lies at the core of multimedia. Although many technologies are involved, video coding and its standardisation are certainly key enablers of these developments. Video compression has always been an important area of research due to practical limitations on the amount of information that can be stored, processed, or transmitted.

This thesis addresses the computational complexity problem of software video encoders. In this work, a number of novel techniques aimed at reducing the computational cost of ME process are presented. In particular, the optimisation of the interpolation effect on ME process is one of the main contributions of this thesis. This involves the interpolation effect on the spatial domain and the frequency domain ME. Algorithms that exploit this effect have been proposed and presented, including accurate initial prediction, dynamic search range, simplified search flow and intelligent early termination. Such approaches offer not only significant computational time savings but also flexibility to an encoder for heterogeneous applications working towards complexity-distortion optimality. All the proposed methods have been integrated and tested versus the latest H.264/AVC, SVC and MVC schemes.

This chapter summarises the main contributions of this work. Initially, the completion of each stage of the project towards the final objective is highlighted while emphasising the relevance of this work to the research problem. Then the algorithms and experimental results are critically reviewed. The advantages and disadvantages of the main contributions are discussed. Finally, possible directions for further research in relation to the main findings are also indicated.

9.2 Outcome

9.2.1 Overview

The H.264 video coding standards can deliver significantly improved compression efficiency and coding flexibility compared to other video coding standards. Due to increased compression efficiency and coding flexibility, H.264/AVC has the potential to enable new video services such as mobile video telephony and multimedia streaming over mobile networks. However, the performance gains of H.264/AVC come at a cost of significantly increased computational complexity and therefore the processing resources required to implement an H.264/AVC encoder in a power-constrained mobile platform are likely to be a major problem. The objective of this research project was to develop novel algorithms to manage the computational complexity of an H.264 encoder. These algorithms required to effectively utilise the available computational resources to maximise the rate constrained video quality of the encoder.

The research project has been structured into three stages as illustrated in chapter 1. Every stage has been successfully completed. The work carried out and the algorithms developed have been presented in this thesis.

9.2.2 Stage 1

This stage of the project involved evaluating the performance of the H.264/AVC encoder with different coding parameters and identifying the main contributors to computational complexity. It was established that the ME and mode decision process consumes the largest proportion of the processing resources during encoding. Additionally literature review has been carried out to identify the advantages and disadvantages of proposed complexity reduction algorithms.

In chapter 2 an introduction to the different concepts and techniques used in video compression were presented. The advantages and disadvantages of state of the art video coding standards were illustrated by giving a detailed description of the ME and mode decisions process which are considered to be the main root of the advance of the H.264 standard when compared to previous standards, and the main area of this thesis novel contribution.

Chapter 3 presented the experimental methods used to evaluate video coding algorithms. The chapter presents the three main characteristics for the measurement to assess the performance of video compression algorithms. Those are compression ratio, quality and compression speed, also the rate-distortion theory and this thesis specific experimental setup were outlined in this chapter. In this thesis the performance of the algorithms is evaluated by measuring the computational complexity, bit rate and objective video quality, to ensure equal comparison between the proposed algorithms against the standard and recently developed algorithms.

The implications of exploiting different distortion metrics in the ME different layers are discussed in chapter 5. Moreover, a comparison between the SAD and SATD effect on the coefficients bits and motion vector bits on different layers were presented and analysed. All the remaining chapters, where novel algorithms were proposed, have started with a critical review of existing low-complexity algorithms in the corresponding area.

9.2.3 Stage 2

In this stage new complexity reduction algorithms for H.264/AVC encoder were developed. Particularly, two novel algorithms were presented to speed up the ME and the MD process in the H.264/AVC. In the first one, an existing fast frequency domain ME algorithm [13] was identified as a suitable candidate to enhance the H.264/AVC ME and MD process. Computation savings has been achieved because some of the processing of the MBs predicted as skipped or predicted using large modes can be avoided and therefore the computational resources needed for ME and subsequent processing for these MBs can be saved. This algorithm was applied to an H.264/AVC reference encoder in chapter 4. Experimental results showed that significant amount of computational complexity can be saved with a small loss of rate-distortion performance by carefully selecting the thresholds that control the amount of MBs predicted as skipped or predicted using large modes.

The second algorithm was presented in chapter 6. In the chapter a novel sub-pixel ME was presented. The algorithm is based on statistical analysis that has been carried out to define situations when sub-pixel ME is redundant for the different MBs partitions. The results indicated that the computational complexity of the encoding process is reduced by 11-62% with negligible loss of rate-distortion performance.

9.2.4 Stage 3

This stage focused on optimising video encoders for nowadays video applications where multiple video streams with different characteristics are coded simultaneously to enhance the end-user experience. With an aim to speed up the realisation of those applications in real time, in this stage, the complexity reduction algorithms proposed in chapter 6 was further developed and applied to the Scalable and Multi view video coding extensions, chapter 7 and 8 respectively. This has been done in order to enable control of the additional computational complexity added to those extensions. The results show that effective and scalable control of complexity is achieved for both the SVC and the MVC with minimum loss in rate-distortion performance.

9.3 Original Contribution

This research project has contributed to the design of fast and simple ME and MD methods that can be adopted into the H.264/AVC, SVC and MVC standards to improve the coding efficiency, the contributions resulted in novel algorithms for three major areas of digital video compression; single stream video, scalable videos and multi-view videos.

Additionally, throughout the process, the research provided the body of knowledge with original contribution that includes an in-depth analysis and evaluation of different aspects of the currently proposed and implemented methods.

The contributions of this work which are considered to be novel are as follows:

9.3.1 The effect of the SATD on motion estimation

Since ultimately the transformed coefficients are coded, better estimation of the cost by can be achieved by estimating the effect of the DCT with a 4×4 Hadamard transform. Although these advantages are well known and the Hadamard transform is implemented in various parts of the ME and MD processes of the standard, little research has been carried out to investigate the effect of the λ selection and the interpolation on the SATD. The reason for this is the heavy computational that required to execute the SATD; which involves subtraction, addition, shift and absolute operations. However, if the ME is improved and the SATD is selectively applied as indicated in [74], in addition to enhancing the effect of the DCT, significant RD enhancement can be achieved in wide range of applications. Particularly, in hardware applications when applying the same distortion metric at different resolutions is essential.

9.3.2 Frequency Domain Motion Estimation

The objective here was the introduction of frequency domain motion estimation to the standard ME and MD processes. The advantages of the phase correlation ME when compared to the block matching techniques are the computational efficiency and the generation of true motion. Therefore, the phase correlation is used to generate true motion vector in applications such as video surveillance and moving object detection. However, in comparison to the block matching techniques the phase correlation results in higher bit rate. In this work a modification of the standard block matching technique to include a pre-processing implementation of the phase correlation are proposed. Moreover, the correlation of true motion is used to adjust the ME and MD processes. This results in significant time saving.

9.3.3 Fast sub-pixel Motion Estimation algorithms

In order to further improve the prediction accuracy and thus the compression efficiency, the concept of non-separable adaptive interpolation filters that are specifically adapted to statistics of the current image are introduced in [93, 123]. Although those techniques successfully enhanced the coding efficiency, complexity of these schemes was analyzed in [124] and it was shown that compared to the standard H.264/AVC interpolation, these schemes have approximately up to 3 times more decoding complexity. In contrast, no research has been carried out to investigate and define situations when the interpolation process is unnecessary as proposed in this work [75, 76].

Further advantages of the proposed work include; its simple implementation in hardware and software and its clear possible development to include MC with 1/8-pel motion vectors [93]. The latter can be accomplished by adding an extra step to the proposed algorithm. Moreover, the proposed algorithm can be applied to the recently developed separable and non-separable adaptive interpolation filters.

9.3.4 Scalable and Multiview Video Coding

The strongest contributions of the thesis are fast hierarchal prediction schemes for the H.264 extensions for scalable and multiview video coding [94, 95, 108]. Multiview in particular is highly relevant topic to the current research being undertaken by number of institutions attempting to advance real time multiview application.

H.264/SVC caters to requirements of wide applications by eliminating the need for transcoding to support multiple resolutions. In SVC encoder and decoder, all resolutions from QCIF to full-HD can be derived from a single video stream, thereby enabling companies to develop embedded systems at lower costs and faster time-to-market.
However, the standard has unbearable complexities for real-time encoding. Therefore, there is a tremendous need to reduce encoding complexity and to design a flexible, rate distortion optimised, yet computationally efficient encoder for various applications. Adaptive error-resilient motion estimation with optimal complexity-rate-distortion is vital for future wireless low power video applications. Thus, optimising the H264/SVC ME and MD has been considered to be one of the main novel contributions of this project.

A natural continuation of the work on SVC was to further develop the proposed algorithm for applications of the H.264/MVC multiview extension. Resulting in a novel proposed algorithm for efficient MVC. The algorithm achieves significant time saving when compared to the standard reference software and recently reported algorithms. It makes use of the fast sub-pixel ME and fast hierarchal prediction scheme proposed for the H.264/SVC to speed up ME time in the MVC. In addition to terminating the sub-pixel ME and selecting the mode adaptively, an extra step has been added to control the references for any MB in any view.

In comparison to recently proposed methods, the algorithm is unique as it does not take advantage of the spatial mode distribution between MBs, instead it relays on the relationship between the MB and its enclosed partitions. The advantage of this in contrast to other schemes is the obvious consistency of the resultant RD performance.

9.3.5 Summary

The methods proposed in this work are likely to be particularly useful for encoders implemented on low power handheld devices. Example applications include:

1. Switching to a low power encoding mode to prolong the battery life while maximising the perceptual video quality given the selected power level.

2. Maintaining perceptual quality of encoding while more power is diverted to background activities such as transmission (in a mobile video phone) to combat a sudden signal drop or interference.

Furthermore, the newly proposed Scalable and Multiview coding schemes can significantly contribute to the current ongoing research to advance real-time multimedia applications that aimed at enhancing the end-user experience. Examples of those applications include 3D multi-view TV and Free Viewpoint television.

9.4 Future Work

The algorithms developed during this research project were summarised and critically evaluated in the previous section. This section presents some directions for further research, mainly aimed at extending the algorithms to achieve better performance and flexibility.

The algorithms presented so far are evaluated using fixed quantisation parameters values. Experiments should be carried out to evaluate the performance of the low complexity algorithms combined with established rate control methods. This will highlight compatibility/interoperability issues of combined rate and complexity control based on the above algorithms and may open further avenues for investigation.

From the conclusion of chapter 5, further research needs to be carried out to define more appropriate interpolation filters that can accommodate the exploitation of the Hadamard transform in full-pixel ME. Furthermore, the lagrange multiplier needs to be retrained or be dynamically adaptable to optimise the ME process for different distortion metrics.

In chapter 4, the phase correlation approach was used as a pre-processing step for the ME and MD process. This can be further extended and applied to the MVC on frame level as a pre-processing step to early terminate the ineffective reference frames.

Bibliography

[1] I. E. G. Richardson. *H.264 and MPEG-4 Video Compression*. John Wiley & Sons Ltd, 2003.

[2] I. E. G. Richardson and M. J. Riley. "ATM Cell Loss Effects on a Progressive JPEG Video Codec." in *Proc. 3rd International Conference on Broadband Islands*, June 1994, pp.155-165.

[3] "International Telecommunications Union." Internet: http://www.itu.int/ITU-T/, [February 2011]

[4] "International Orgranization for Standardization." Internet: http://www.iso.org/, [February 2011]

[5] *"International Electrotechnical Commission." Internet:* http://www.iec.ch/, [February 2011]

[6] ITU-T: Line Transmission on Non-telephone Signals: Video Codec for Audiovisual Services at px64 kbit/s. CCITT Recommendation H.261,1990.

[7] MPEG-1: International Standard ISO/IEC IS 11172. Coding of Moving Pictures and Associated Audio for Digital Storage Media up to about 1.5Mbit/s,1993.

[8] MPEG-2: International Standard ISO/IEC IS 13818, Generic coding of moving pictures and associated audio Information, 1995.

[9] MPEG-4: International Standard ISO/IEC JTC1/SC29/WG11 N4030, Coding of Moving Pictures Audio, 2001.

[10] T. Wiegand, G.J. Sullivan, G. Bjontegaard, and A.Luthra. "Overview of the H.264/AVC Video Coding Standard" *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, PP. 560-576, Jul. 2003.

[11] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockammer and T. Wedi. "Video Coding with H.264/AVC: Tools, Performance, and Complexity." *IEEE Circuits and Systems*, vol. 4, no. 1, pp. 7-21, April 2004.

[12] T. Wedi and H. G. Musmann. "Motion- and Aliasing-Compensated Prediction for Hybrid Video Coding." *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 577-586, Jul. 2003.

[13] J. J. Pearson, D. C. Hines, S. Goldman, and C. D. Kuglin. "Video rate image correlation processor," in *Proc. SPIE, Application of Digital Image Processing*, vol. 119, pp 197-205, August 1977./

[14] H. Schwarz, D. Marpe, and T. Wiegand. "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103-1120, Sep. 2007.

[15] A. Smolic. "Introduction to Multiview Video Coding", ISO/IEC JTC 1/SC 29/WG 11N9580 , Antalya, Turkey, Jan. 2008.

[16] S. Westland. “Models of the visual system and their application to image-quality assessment," presented at AIC Colour 05 - 10th Congress of the International Colour Association, Spain, pp. 8-13 May 2005.

[17] R. Hoffman *Data compression in digital Systems*. NewYork: Chapman & Hall, 1997.

[18] G. H. Barry, P. Atul and N. N. Arun. *Digital video: an introduction to MPEG-2*. New York: Chapman & Hall.

[19] A. Rosenfeld. *Picture Processing by Computer* .New York: Academic Press, 1969.

[20] David A. COOK *A history of narrative film* 2nd Edition, p2 W.W. Norton & Compnay Inc. New York, 1990 ISBN: 0-393-95553-2

[21] V. Bhaskaran and K. Konstantinides. *Image and Video Compression Standards - Algoritms and Architectures*. Boston, MA: Kluwer Academic Publishers, 1997.

[22] ITU-T. Video codec for audiovisual services at p×64 kbits/s, 1993. ITU-T Recommendation H.261, Version 2.

[23] ITU-T. Video coding for low bit rate communications, 1998. ITU-T Recommendation H.263, Version 2.

[24] ITU-T and ISO/IEC JTC1. Generic coding of moving pictures and associated audio information – Part 2: Video, 1994. ITU-T recommendation H.262 and ISO/IEC 13818-2 (MPEG-2).

[25] ISO/IEC. Coding of audiovisual objects – Part 2: Visual, 1999. ISO/IEC 14496-2 (MPEG-4).

[26] ITU-T Rec. H.264 and ISO/IEC 14496-10 (MPEG-4 AVC), ITU-T and ISO/IEC JTC 1. “Advanced Video Coding for Generic Audiovisual Services”, Version 1: May 2003, Version 2: May 2004, Version 3: Mar. 2005, Version 4: Sept. 2005, Version 5 and Version 6: June 2006, Version 7: Apr. 2007, Version 8 (including SVC extension): November 2007.

[27] G. Bjontegarrd and K. Lillevold. “Context-adaptive VLC coding of coefficients.” in *JVT Document*, vol. JVT-C028, Fairfax, Virginia, USA, May 2002.

[28] D. Marpe, H. Schwarz, G. Bldttermann, G. Heising, and T. Wiegand,. “Context-based Adaptive Binary Arithmetic Coding in JVT/H.26L" in *Proceedings of International Conference on Image Processing*, Rochester, NY, USA, pp. 513-516, 2001.

[29] H. Jong, L. Chen, and T. Chiueh. “Parallel Architectures for 3-step Hierarchical Search Block-matching Algorithm.” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 4, no. 4, pp. 407-416, Aug. 1994.

[30] K. P. Lim, G. Sullivan, and T. Wiegand “Text Description of Joint Model Reference Encoding Methods and Decoding Concealment Methods.” *Document JVT-N046, Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG*, Hong Kong, Jan. 2005.

[31] Karsten Sühring. “H.264/AVC Reference Software Version” http://iphome.hhi.de/suehring/tml/download/, Joint Video Team,2003.

[32] “Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H.264/ISO/IEC 14 496-10 AVC,” in Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVTG050, 2003.

[33] G. Cote and L. Winger. “Recent Advances in Video Compression Standards.” *IEEE Canadian Review*, pp. 21-24, 2002.

[34] VCEG, Joint Video Team of ISO IES MPEG and ITU-T. Joint Committee Draft (CD) of Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC). T. Wiegand, 2002.

[35] G.Sullivan, P.Topiwala and A.Luthra. “The H.264/AVC Advanced Video Coding Standard: Overview and Introduction to the Fidelity Range Extensions.” *SPIE Conference on Applications of Digital Image Processing XXVII*, vol. 5558, pp. 53-74, Aug. 2004.

[36] D.Marpe, T.Wiegand and S.Gordon. “H.264/ MPEG-4 AVC Fidelity Range Extensions: Tools, Profiles, Performance and Application Areas” in *Proc. ICIP*, September 11-14, 2005, Genova, Italy.

[37] M. E. Al-Mualla, C. N. Canagarajah and D. R. Bull. *Video Coding for Mobile Communications, Efficiency, Complexity, and Resilience*. California: Academic Press, 2002.

[38] Z. Wang, L. Lu and A. C. Bovik. "Video Quality Assessment Based on Structural Distortion Measurement." *Signal Processing: Image Communications*, vol. 19, no. 1, pp.. 1-9. Jan. 2004.

[39] C. E. Shannon. "A Mathematical Theory of Communication." *Bell System Technical Journal*, vol. 27, pp. 379-423,623-656, Jul., Oct. 1948.

[40] “Intel VTune Amplifier XE and VTune Performance Analyzer.” Internet: http://software.intel.com/en-us/forums/intel-vtune-performance-analyzer/. [February 2011]

[41] ISO/IEC MPEG and ITU-T VCEG Joint Video Team. "JVT-Q042, Revised H.264/MPEG-4 AVC Reference Software Manual," *17th Meeting*: Nice, FR, 14-21 October 2005.

[42] J. Reichel, H. Schwarz, and M. Wien, J. Reichel, and M. Wien, ”Joint Scalable Video Model 9.15 (JSVM 9.15),” Joint Video Team, Doc. JVT-V202, March 2009.

[43] ISO/IEC JTC1/SC29/WG11. Joint Multiview Video Model (JMVM) 7.0., JVT Document, JVT-Z207, Jan. 2008.

[44] Y. Su, A. Vetro, and A. Smolic. “Common Test Conditions for Multiview Video Coding” *ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6)*, Document: JVT-T207, , pp. 15–21, 2006.

[45] G. Bjontegaard, “Recommended Simulation Conditions for H.26L”, *ITU-T SG16/Q15*, doc. Q15-I-62, Red Bank, Oct. 99.

[46] G. Bjøntegaard. “Calculation of average PSNR differences between RD-Curves,” *ITU-T SG16 Q.6 Document*, VCEG-M33, Austin TX, USA, April 2001.

[47] A. Abdelazim, S. Mein, M. Varley, C. Grecos and D. Ait-Boudaoud "Phase correlation based adaptive mode decision for the H.264/AVC", *SPIE Electronic Imaging Conference*, San Francisco CA , 23-27 January 2011.

[48] A. Abdelazim, S. Mein, M. Varley and D. Ait-Boudaoud "Fast Mode Decision for the H.264/AVC Based on Frequency Domain Motion Estimation", *Optical Engineering Letters*, ISSN 0091-3286, vol. 50, Issue 7, July 2011.

[49] C. Grecos, M. Y. Yang, V. Argiriou, P. Lambert, J. Slowak, S. Mys, J. Skorupa and R. W. De Walle. "Fast Mode Decision in H.264/AVC," in *Handheld Computing for Mobile Commerce: Applications, Concepts and Technologies*, IGI Global, 2010.

[50] J. L. Mitchell, W. B. Pennebaker, C. E. Fogg, and D. J. LeGall. *MPEG Video Compression Standard*. Boston, MA: Kluwer Academic Publishers, 1996.

[51] A. M. Tourapis, Wu Feng and Li Shipeng. "Direct Mode Coding for Bipredictive Slices in the H.264 Standard." *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 1, Jan. 2005

[52] Y. Zhao and I. E. G. Richardson. "Macroblock Classifications for Complexity Management of Video Encoders." *Signal Processing: Image Communincation*, vol. 18, no. 9, pp. 801–811, Oct. 2003.

[53] C. Grecos & M. Y. Yang. "Fast Inter Mode Prediction for P Slices in the H264 Video Coding Standard." *IEEE Transaction on Broadcasting*, vol. 51, issue 2, Jun. 2005.

[54] A. Chia, Woo Yu, Graham R. Martin & Heechan Park. "Fast Inter-Mode Selection in the H.264/AVC Standard Using a Hierarchical Decision Process." *IEEE Transaction on Circuits and System for Video Technology*, vol. 18, no. 2, Feb. 2008.

[55] B. Meng, O. C. Au, C. Wong and H. Lam. "Efficient Intra-prediction Mode Selection for 4x4 Blocks in H.264." *2003 IEEE lot. Conf. Multimedia &Expo (ICME2003),* Baltimore. MD, USA, Jul. 2003.

[56] C. S. Kannangara, I. E. G. Richardson, M. Bystrom, J. Solera, Y. Zhao, A. MacLennan and R. Cooney. "Low Complexity Skip Prediction for H.264 Through Lagrangian Cost Estimation." *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 2, pp. 202-208, Feb. 2006.

[57] B. Jeon and J. Lee. "Fast Mode Decision for H264." *ISO/IEC JTC1/SC29/WG11 and ITU-T SG16*, Input Document JVT-J033, Dec. 2003.

[58] K. P. Lim, S.Wu, D. J. Wu, S. Rahardja, X. Lin, F. Pan and Z. G. Li, "Fast Inter Mode Selection", *ISO/IEC JTC1/SC29/WG11 and ITU-T SG16*, Input Document JVT-I020, Sept. 2003.

[59] X. Jing and L.-P. Chau. "Fast Approach for H.264 Inter-mode Decision." *Electronic Letters*, vol. 40, no. 17, pp. 1050-1052, Sep. 2004.

[60] A. Chang, O.C. Au and Y.M. Yeung. "A Novel Approach to Fast Multi-block Motion Estimation for H.264 Video Coding," *International Conference on Multimedia and Expo, 2003. ICME'03*, vol. 1, pp. 6-9, Jul. 2003.

[61] H. Kim and Y. Altunbasak. "Low-complexity Macroblock Mode Selection for H.264/AVC Encoders." *IEEE International Conference on Image Processing, 2004, ICIP'04*, vol. 2, pp. 24-27, Oct. 2004.

[62] A. Tanizawa, S. Koto, T. Chujoh, & Y. Kikuchi. "A Study on Fast Rate-distortion Optimized Coding Mode Decision for H.264," *IEEE International Conference on Image Processing, 2004, ICIP'04*, vol. 2, pp. 24-27, Oct. 2004.

[63] A. Ahmad, N. Khan, S. Masud and M.A. Maud. "Efficient Block Size Selection in H.264 Video Coding Standard," *Electronics Letters*, vol 40, no. 1, Jan. 2004.

[64] Zhou Z. and M.T. Sun, "Fast Macroblock Inter Mode Decision and Motion Estimation for H.264/MPEG-4 AVC," *IEEE International Conference on Image Processing, ICIP'04*. vol. 2, pp. 24-27, Oct. 2004.

[65] L. I. Kuncheva. *Fuzzy Classifier Design*. New York: Springer, 2000.

[66] M. Yang and W. Wang. "Fast Macroblock Mode Selection Based on Motion Content Classification in H.264/AVC," *IEEE International Conference on Image Processing*, *ICIP'04*, vol. 2, pp. 24-27, Oct. 2004.

[67] J. Seok, J. W. Lee and C. S. Cho. "Fast Block Mode Decision Algorithm in H.264/AVC using a Filter Bank of Kalman Filters for High Definition Encoding," *in Multimedia Systems*, vol. 13, no. 5-6, Feb. 2008.

[68] G. A. Thomas. "Television Motion Measurement for DATV and Other Applications." *BBC Res. Dept. Rep.*, no. 1987.

[69] Y. Ismail, M. Shaaban, and M. Bayoumi. "An Adaptive Block Size Phase Correlation Motion Estimation Using Adaptive Early Search Termination Technique." *IEEE International Symposium on Circuits and Systems*, *ISCAS 2007*, pp. 3423–3426, May 2007.

[70] M. Paul and G. Sorwar. "An Efficient Video Coding using Phase-matched Error from Phase Correlation Information." *IEEE 10th Workshop on Multimedia Signal Processing*, pp. 378-382, Oct. 2008.

[71] C. Stiller, J. Konrad. "Estimating Motion in Image Sequences", *IEEE Signal Processing Magazine*,vol 16, no. 4, pp. 70-91, 1999.

[72] X. Li, N. Oertel, A. Hutter and A. Kaup. 2009. "Laplace Distribution Based Lagrangian Rate Distortion Optimization for Hybrid Video Coding", *IEEE Transactions on Circuits and Systems for Video Technology*, 19(2), pp. 193-205, 2009.

[73] T. Wiegand, and B. Girod. "Lagrange Multiplier Selection in Hybrid Video Coder Control." *IEEE International Conference on Image Processing, (ICIP)*, Thessaloniki, Greece, vol. 3, pp. 542-545, 2001.

[74] A. Abdelazim, M. Varley and D. Ait-Boudaoud. "Effect Of The Hadamard Transform on Motion Estimation of Different Layers in Video Coding." *Close Range Image Measurement Techniques*, Newcastle upon Tyne, 22-24 Jun. 2010.

[75] A. Abdelazim, M. Y. Yang, C. Grecos and D. Ait-Boudaoud. "Selective Application of Sub-pixel Motion Estimation and Hadamard Transform in H264 AVC", *SPIE Electronic Imaging Conference*, San Jose CA, USA, 18-22 Jan. 2009.

[76] A. Abdelazim, M. Yuan Yang and C. Grecos. "Fast Sub-Pixel Motion Estimation Based on the Interpolation Effect on Different Block Sizes for H264/AVC", *Optical Engineering Letters*, ISSN 0091-3286, vol. 48, issue 3, Mar. 2009.

[77] Z. B. Chen, P. Zhou and Y. He. "Fast Integer Pel and Fractional Pel Motion Estimation for JVT," *JVT-F017,* 6th meeting: Awaji, Japan,pp. 5-13 Dec. 2002.

[78].A. Tourapis, "Enhanced Predictive Zonal Search for Single and Multiple Frame Motion Estimation," in *Proc. of VCIP 2002*, pp.1069-79, Jan. 2002.

[79] Z. B. Chen, C. Du, J. H. Wang and Y. He. "PPFPS – A Paraboloid Prediction Based Fractional Pixel Search Strategy for H.26L," in *Proc. of ISCAS 2002*, pp. 9-12, May 2002.

[80] Z. B. Chen and Y. He. "Prediction Based Directional Refinement (PDR) Algorithm for Fractional Pixel Motion Search Strategy," *JVT-D069*, 4th meeting: Klagenfurt, Austria, pp. 22-26 Jul. 2002.

[81] H. M. Wong, O. C. Au and A. Chang. "Fast Sub-Pixel Inter-Prediction - Based on the Texture Direction Analysis," in *Proc.IEEE ISCAS*, 2005, vol. 6, pp. 5477-5480.

[82] Y.-J. Wang, C.-C. Cheng and T.-S Chang. "A Fast Algorithm and its VLSI Architecture for Fractional Motion Estimation for H.264/MPEG-4 AVC Video Coding," *IEEE Transactions on Circuits And Systems for Video Technology*, vol. 17, no. 5, pp. 578-583, May 2007.

[83] P. R. Hill, T. K. Chiew, D. R. Bull, and C. N. Canagarajah. "Interpolation Free Subpixel Accuracy Motion Estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no.12, pp. 11519-26, Dec. 2006.

[84] Jung W. Suh and Jechang Jeong. "Fast Sub-pixel Motion Estimation Techniques Having Lower Computational Complexity," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 3, Aug. 2004.

[85] Yu-Jen Wang, Chao-Chung Cheng, Tian-Sheuan Chang. "A Fast Fractional Pel Motion Estimation Algorithm for H.264/MPEG-4 AVC", *Proc. IEEE International Symposium on Circuits and Systems (ISCAS'2006)*, May 2006, Island of Kos, Greece. pp. 3974-7.

[86] C. Du, Y. He, and J. Zheng. "PPHPS: A Parabolic Prediction-Based, Fast Half-Pixel Search Algorithm for Very Low Bit-Rate Moving-Picture Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 6, pp. 514-518, Jun. 2003.

[87] J.W. Suh, J. Jeong. "Fast Sub-pixel Motion Estimation Techniques Having Lower Computational Complexity," *IEEE Transactions on Consumer Electronic*, vol. 50, pp. 968-973, Aug. 2004.

[88] K. Chao-Yang, K. Huang-Chih, and L. Youn-Long. "High Performance Fractional Motion Estimation and Mode Decision for H.264/AVC," in *Proc. IEEE ICME*, 2006, pp. 1241-1244.

[89] M. Sayed, W. Badawy, and G. Jullien "Low-Complexity Algorithm For Fractional-Pixel Motion Estimation" in *Proc. of ICIP 2009*, pp. 1565-1568, Sep. 2009.

[90] Y. Zhang, Wan-C. Siu and T. Shen. "Fast Sub-pixel Motion Estimation Based on Directional Information and Adaptive Block Classification" in *Proc. of VIE 2008*, pp. 622-627, 2008.

[91] P. R. Hill and D. R. Bull. "Kernel Based Sub-pixel Motion Estimation" in *Proc. of ICIP 2009*, pp. 1557-1560, Sep. 2009.

[92] D. Rusanovskyy, K. Ugur, and J. Lainema. “Adaptive Interpolation with Directional Filters.” *Video Coding Experts Group (VCEG) 33rd Meeting*: Shenzhen, China, 20 Oct. 2007.

[93] J. Ostermann, and M. Narroschke. “Motion Compensated Prediction with 1/8-pel Displacement Vector Resolution.” *Video Coding Experts Group (VCEG) 30th Meeting*: Hangzhou, China, 23-27 Oct. 2006.

[94] A. Abdelazim, S. Mein, M. Varley and D. Ait-Boudaoud. “Low Complexity Hierarchical Prediction Algorithm for H.264/SVC,” *The Fourth Pacific-Rim Symposium on Image and Video Technology*, Singapore , 14-17 Nov. 2010.

[95] A. Abdelazim, S. Mein, M. Varley, C. Grecos and D. Ait-Boudaoud. “Fast Multilayered Prediction Algorithm for Group of Pictures in H.264/SVC,” *SPIE Electronic Imaging Conference*, San Francisco CA, USA, 23-27 Jan. 2011.

[96] H. Li, Z. Li and C. Wen. “Fast Mode Decision Algorithm for Inter-Frame Coding in Fully Scalable Video Coding,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 7, pp. 889-895, Jul. 2006.

[97] G. Goh, J. Kang, M.Cho and K. Chung. “Fast Mode Decision for Scalable Video Coding Based on Neighboring Macroblock Analysis.” *ACM Symposium on Applied Computing*, pp. 1845 ~ 1846, 2009.

[98] J. Ren and N. Kehtarnavaz. “Fast Adaptive Early Termination for Mode Selection in H.264 Scalable Video Coding.” *IEEE International Conference on Image Processing (ICIP’08)*, San Diego, CA, USA, Oct. 2008.

[99] Y. Dewier, B. Hou, and R. Sotudeh. “A Fast Mode Decision Algorithm in Spatial and Temporal Scalable Video Coding,” in *Proc. Third International Symposium on Intelligent Information Technology Application*, 2009, pp. 716-719.

[100] H. C. Lin, H. M. Hang and W. H. Peng “Fast Temporal Prediction for H.264/AVC Scalable Video Coding.” *IEEE International Conference on Image Processing (ICIP’09)*, Cairo, Egypt, pp. 3425-3428, Oct. 2009.

[101] K. Müller, P. Merkle, Al. Smolic, and T. Wiegand,”Multiview Coding using AVC,” MPEG Meeting - ISO/IEC JTC1/SC29/WG11, Bangkok, Thailand, MPEG06/M12945, Jan. 2006.

[102] Jiangbo Lu, Hua Cai, Jian-Guang Lou and Juang Li, “An Epipolar Geometry-based Fast Disparity Estimation Algorithm for Multi-view Image and Video Coding.” *IEEE Tranactions on Circuits and Systems for Video Technology*, vol.17, no.6, pp.737-750, Jun. 2007.

[103] Gangyi Jiang, Mei Yu, Feng Shao, You Yang and Haitao Dong. “Fast Multi-view Disparity Estimation for Multi-view Video Systems.” *LNCS 4179, ACIVS 2006*, 2006, pp. 493-500.

[104] X. Li, D. Zhao, X. Ji, Q. Wang, and W. Gao, “A Fast Inter Frame Prediction Algorithm For Multi-View Video Coding,” in *Proc. ICIP*, San Antonio TX, USA, pp. 417-42, Sept. 2007.

[105] L. Shen, T. Yan, Z. Liu, Z. Zhang, P. An, and L. Yang, "Fast Mode Decision for Multiview Video Coding," in *Proc. ICIP*, Cairo, Egypt, pp. 2953-56, Sept. 2010.

[106] L. Shen, T. Yan, Z. Liu, Z. Zhang, and P. An, "Selective Disparity Estimation and Variable Size Motion Estimation Based on Motion Homogeneity for Multi-View Coding." *IEEE Trans. on Broadcasting*., *Video Technol.*, pp. 55(4):761-766, 2009.

[107] S. Y. Lee, K. M. Shin, and K. M. Shin, "An Object-based Mode Decision Algorithm for Multi-view Video Coding", in *Proc. ISM*, California, USA, 74-81, Dec. 2008.

[108] A. Abdelazim, G. Y. Zhang, St. Mein, M. Varley and D. Ait-Boudaoud. "Fast motion prediction algorithm for multiview video coding", *SPIE Defense, Security, and Sensing Conference*, Orlando, FL, USA, 25-29 Apr. 2011.

[109] Yo Sung Ho, and Kwan Jung Oh. "Overview multi-view video coding" in the Proceedings of Systems *Signals and Image Processing, 2007 and 6th EURASIP Conference focused on Speech and Image Processing, Multimedia Communications and Services*, pp. 5-12, Jun. 2007.

[110] ISO/IEC JTC1/SC29/WG11. Requirements on Multiview Video Coding v.4., Doc. N7282, Poznan, Poland, July 2005.

[111] P. Merkle, A. Smolic, K. Mueller, T. Wiegand. "Efficient prediction structures for Multi-view Video Coding." *IEEE Transactions on Circuits and Systems for Video Technology*, 17(11):1461-1473.

[112] P. Merkle, K. Müller, A. Smolic, and T. Wiegand "Efficient Compression Of Multi-View Video Exploiting Inter-View Dependencies Based On H.264/MPEG4-AVC" *ICME*, pp. 1717-20, Jul. 2006.

[113] U. Fecker and A. Kaup. "Statistical Analysis of Temporal and Spatial Block Matching Algorithm for Multi-view Video Sequences." *ISO/IEC JTC1/SC29/WG11, MPEG2005/M11546*, Jan. 2005, Hong Kong, China.

[114] ISO/IEC JTC1/SC29/WG11, "Survey of Algorithms used for Multi-view Video Coding (MVC)", Doc. N6909, Hong Kong, China, January 2005.

[115] A. Smolic, and P. Kauff. "Interactive 3D Video Representation and Coding Technologies," in *Proc. of the IEEE, Special Issue on Advances in Video Coding and Delivery*, vol. 93, no. 1, Jan. 2005.

[116] K.-J. Oh, and Y.-S. Ho. "Multi-view Video Coding based on the Lattice-like Pyramid GOP Structure," in *Proc. PCS 2006, Picture Coding Symposium*, Beijing, China, Apr. 2006.

[117] X. Cheng, L. Sun, and S. Yang, "A Multi-view Video Coding Scheme Using Shared Key Frames for High Interactive Application," in *Proc. PCS 2006, Picture Coding Symposium*, Beijing, China, Apr. 2006.

[118]Y. Yang, G. Jiang, M. Yu, F. Li, and Y. Kim, "Hyper-Space Based Multiview Video Coding Scheme for Free Viewpoint Television," in *Proc. PCS 2006, Picture Coding Symposium*, Beijing, China, Apr. 2006.

[119] F. Shao, G. Jiang, M. Yu, and X. Chen, "A New Image Correction Method for Multiview Video System." *ICME 2006, IEEE International Conference on Multimedia and Expo*, Toronto, Ontario, Canada, Jul. 2006.

[120] A. Kaup and U. Fecker, "Analysis of Multi-Reference Block Matching for Multi-View Video Coding." in *Proc. 7th Workshop Digital Broadcasting*, pp. 33-39, Erlangen, Germany, Sep. 2006.

[121] H. Schwarz, D. Marpe, and T. Wiegand. "Analysis of Hierarchical B Pictures and MCTF." *ICME 2006, IEEE International Conference on Multimedia and Expo*, Toronto, Ontario, Canada, Jul. 2006.

[122] Ping An, Chaohui Lu, Zhaoyang Zhang. "Object Segmentation Using Stereo Images," *ICCCAS*, vol. 1, pp. 534-538, Jun. 2004.

[123] Y. Vatis and J. Ostermann, "Prediction of P- and B-Frames Using a Two-dimensional Non-separable Adaptive Wiener Interpolation Filter for H.264/AVC." *Video Coding Experts Group (VCEG) 30th Meeting*, Hangzhou, China, 23-27 Oct. 2006.

[124] Y.Vatis and J. Ostermann. "Comparison of Complexity Between Two-dimensional Non-separable Adaptive Interpolation Filter and Standard Wiener Filter." *ITU-T SGI 6/Q.6 Doc. VCEG-AA11*, Nice, France, Oct. 2005.

APPENDIX A

MODE DECISION IN H.264/AVC JM SOFTWARE

A.1 Overview

The H.264/AVC standard suggests two mode decision schemes (mutually exclusive) for encoders: a high-complexity mode decision, also known as Rate Distortion Optimised RDO-based mode decision, and a low-complexity mode decision. The mutual exclusion of the two schemes is due to the fact that each contains very different sets of tools for applying the mode decision process. Both schemes are user configurable from the encoder configuration files in the reference software.

A.2 High Complexity Mode Decision for H.264/AVC

In the high complexity mode of the H.264/AVC standard, the optimal MB mode is the one that minimises the following Lagrangian functional:

$$J(o, r, MO \mid QP, \lambda_{MO}) = SSD(o, r, MO \mid QP) + \lambda_{MO} \times R(o, r, MO \mid QP) \qquad \text{(A.1)}$$

In the above equation (A.1), J denotes the cost function which is dependent on o (the original signal block), r (the reconstructed signal block) and MO (selected from a set of modes to be explained shortly). J is found for a given QP (Quantisation Parameter) and λ_{MO} (the Lagrange multiplier for mode decision). The SSD (Sum of the Squared Differences) metric is calculated between the original signal block and its reconstruction and it also depends on the chosen mode (MO). SSD is found for a given QP and for blocks with 4:2:2 sampling ratios using the following equation (A.2):

$$\begin{aligned} SSD(o,r,MO \mid QP) = &\sum_{x=1}^{16}\sum_{y=1}^{16}(o_Y[x,y]-r_Y[x,y,MO \mid QP])^2 \\ &+\sum_{x=1}^{8}\sum_{y=1}^{8}(o_U[x,y]-r_U[x,y,MO \mid QP])^2 \\ &+\sum_{x=1}^{8}\sum_{y=1}^{8}(o_V[x,y]-r_V[x,y,MO \mid QP])^2 \end{aligned} \tag{A.2}$$

Where $r_Y[x,y,MO \mid QP]$ and $o_Y[x,y]$ represent the reconstructed and original luminance values and r_U, r_V, o_U, o_V the corresponding chrominance values. The Lagrange multiplier λ_{MO} [10] depends on the frame type, the number of frames in the sequence and on the quantisation step size per block.

Finally, the rate $R(o,r,MO \mid QP)$ depends on the original and reconstructed block as well as on the chosen mode (MO) for a given QP and reflects the number of bits produced for header(s) (including mode (MO) indicators), motion and texture information. It is worth mentioning that an encoder is free to calculate this rate by either measuring or by estimating it. In the high complexity mode, the reference encoder measures this rate, i.e. it will encode the block up to and including entropy coding. In equation (A.1), MO is chosen from the set of potential prediction modes as follows:

For Intra frames from a set of two modes for 4×4 and 16×16 block sizes:

$$MO \in \{INTRA\_4\times 4, INTRA\_16\times 16\} \tag{A.3}$$

For P frames (single reference frames for forward or backward prediction) from a set of seven modes including both intra and inter modes for various block sizes, as well as the $SKIP$ mode:

$$\begin{aligned} MO \in \{&INTRA\_4\times 4, INTRA\_16\times 16, SKIP, MO\_16\times 16, MO\_16\times 8, \\ &MO\_8\times 16, MO\_8\times 8\} \end{aligned} \tag{A.4}$$

For B frames (bi-directionally predicted framess) from a set of seven modes including both intra and inter modes for various block sizes, as well as the direct mode:

$$\begin{aligned} MO \in \{&INTRA\_4\times 4, INTRA\_16\times 16, DIRECT, MO\_16\times 16, MO\_16\times 8, \\ &MO\_8\times 16, MO\_8\times 8\} \end{aligned} \tag{A.5}$$

The $DIRECT$ mode is particular to the bi-directionally predicted MBs in B slices, while the $SKIP$ mode implies that no motion or texture information is sent to the channel (only the mode (MO) indicator is transmitted).

In the above mode sets, any mode with the prefix $INTRA\_$ will result in encoding the luminance and chrominance error between a directional prediction of a signal block and the block itself. Any mode with the prefix $MO\_$ refers to inter modes of different block sizes. When MO is equal to $INTRA\_4\times4$ or $INTRA\_16\times16$, the best intra mode for each case is chosen through evaluation of equation (A.1) with mode selection from the following sets:

$$\begin{aligned} INTRA\_4\times4 \in \{&dc, horizontal, vertical, diagonal\_down\_right, \\ &diagonal\_down\_left, vertical\_left, vertical\_right, \\ &horizontal\_up, horizontal\_down\} \end{aligned} \tag{A.6}$$

$$INTRA\_16\times16 \in \{dc, horizontal, vertical, plane\} \tag{A.7}$$

A similar functional minimisation results in the choice of the best 8×8 mode for P and B slices from the following set:

$$MO\_8\times8 \in \{inter\_8x8, inter\_8x4, inter\_4x8, inter\_4x4\} \tag{A.8}$$

Any mode with the prefix $MO\_$ in equations (A.4) and (A.5) and inter_ in equation (A.8) assumes that the best motion vector is known for this mode.

Once the best 8×8, $INTRA\_4\times4$, $INTRA\_16\times16$ modes are found, the minimal cost for the MB is evaluated by looping through the different mode possibilities (equations (A.4) and (A.5)).

To comprehend the complexity of the mode decision process, the number of mode evaluations for the luminance and chrominance components (assuming a 4:2:2 resolution) is calculated. For a 16×16 MB (luminance component only) 144 cost evaluations are needed for the best $INTRA\_4\times4$ mode. Adding 4 more evaluations for the $INTRA\_16\times16$ case, 16 more for the best 8×8 inter mode and 7 more for selecting the minimal cost among all modes results in 148 evaluations for MBs in Intra frames and 171 evaluations for MBs in P or B slices. It can also be noted that the two chrominance components of half resolution require approximately the same number of cost evaluations. Given that, in the above analysis the evaluations of the best motion vector calculation that depend on the size of the search window and on the sub-pixel accuracy has not been considered. Clearly this shows that the mode decision process is very computationally intensive.

A.3 Low-Complexity Mode Decision

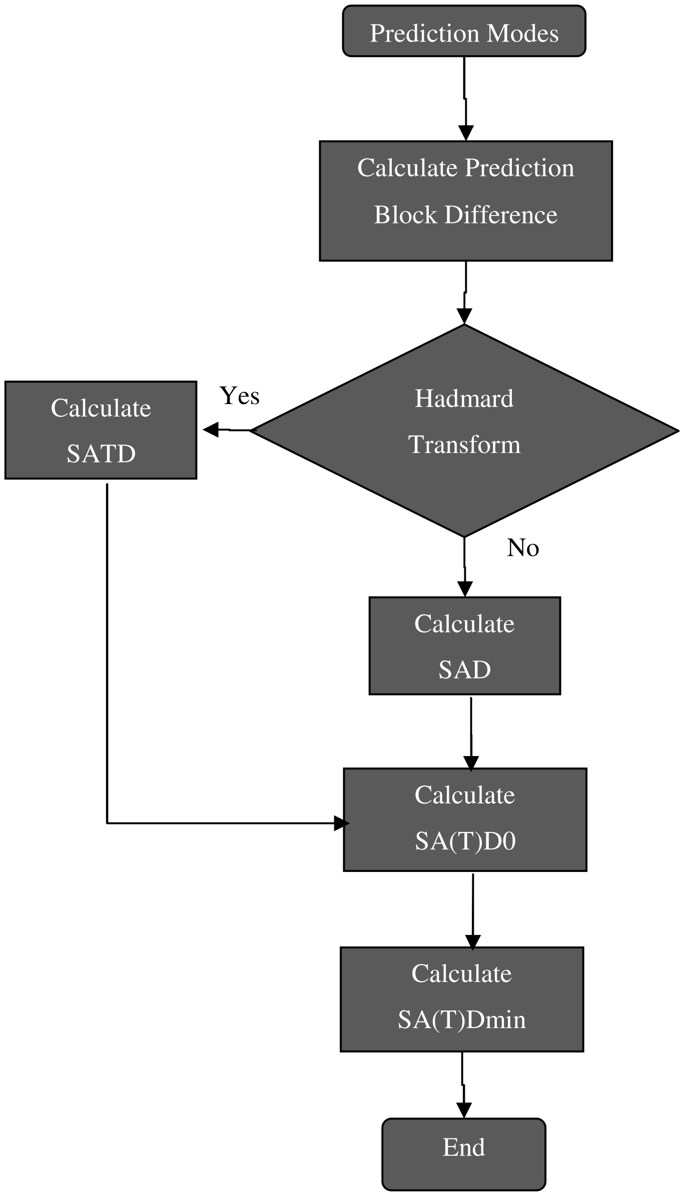


Figure A.1 Flow chart of low complexity mode decision

In the low-complexity mode decision scheme, the cost of each prediction mode is calculated using either the SAD or the $SATD$ of the prediction errors. The cost calculation flowchart is shown in Figure A1.

The $SATD$ to be minimised (last box in the above flow-chart), is given a 'bias' value $SATD_0$ initially in order to favour prediction modes needing a smaller number of bits to be signaled. This bias is the result of the multiplication of a bit usage estimate and a quantization parameter. The bit usage estimate may depend on the number of reference frames (code_number_of_ref_idx_fwd), the rate to signify the chosen block size (Bits_to_code_forward_Blk_size and Bits_to_code_backward_Blk_size) or the rate to code differential motion vectors (Bits_to_code_MVDFW and Bits_to_code_MVDBW) in the equations (A.9) to (A.13). The QP_0 is a table lookup operation with input QP and output QP_0. The calculation of $SATD_0$ at each mode is performed as follows:

Forward prediction mode:

$$\begin{aligned} SA(T)D_0 = QP_0(QP) \times 2(2 \times code\_number\_of\_ref\_idx\_fwd + \\ Bits\_to\_code\_MVDFW) \end{aligned} \tag{A.9}$$

Backward prediction mode:

$$SA(T)D_0 = QP_0(QP) \times Bits\_to\_code\_MVDFW \tag{A.10}$$

Bi-directional prediction mode:

$$\begin{aligned} SA(T)D_0 = QP_0(QP) \times 2(2 \times code\_number\_of\_ref\_idx\_fwd + \\ Bits\_to\_code\_forward\_Blk\_size + \\ Bits\_to\_code\_backward\_Blk\_size + \\ Bits\_to\_code\_MVDFW + \\ Bits\_to\_code\_MVDBW) \end{aligned} \tag{A.11}$$

Direct prediction mode:

$$SA(T)D_0 = -16 \times QP_0(QP) \tag{A.12}$$

Intra 4×4 mode:

$$SA(T)D_0 = 24 \times QP_0(QP) \tag{A.13}$$

Intra 16×16 mode:

$$SA(T)D_0 = 0 \tag{A.14}$$

The $SATD$ is applied on the prediction error block which is defined as the pixel by pixel difference between the original and predicted blocks.

When the $SATD$ path is chosen in Figure A.1, a two dimensional transform is performed on the current block for selecting the best mode. To simplify implementation, the Hadamard transform is chosen.

This transform is performed horizontally and vertically on the error block and results in the Hadamard transformed error block. The $SATD$ between this block and the current block (in a given prediction mode) is given by the equation:

$$SA(T)D = (\sum_{i,j} |DiffT(i,j)|)/2 \tag{A.15}$$

Where $DiffT(i' j)$ is the point by point difference between the 2 blocks. Finally, the minimal cost prediction mode is found by:

$$SA(T)D_{\min} = \min\{SA(T)D + SA(T)D_0)\} \tag{A.16}$$

Low complexity mode decision uses $SATD$ which includes only subtraction and a simple convolution to represent the distortion term. It also uses $SATD_0$ which only includes the table look up computation to find the bit rate term. No DCT, IDCT (Inverse Discrete Cosine Transform) or entropy coding are included in the low complexity mode decision scheme, which implies much lower computation as compared with the high complexity scheme. The average execution time of low-complexity mode decision is only 7% of that of high-complexity mode decision. However, low-complexity mode decision loses an average of 0.48dB in PSNR when compared to the high-complexity mode decision.

APPENDIX B

PUBLICATIONS

Fast subpixel motion estimation based on the interpolation effect on different block sizes for H264/AVC

Abdelrahman Abdelazim, MEMBER SPIE, **Mingyuan Yang,** **and Christos Grecos,** SENIOR MEMBER SPIE
University of Central Lancashire, School of Computing,
Engineering and Physical Sciences, ADSIP
Research Centre, Preston PR1 2HE United Kingdom
E-mail: CGrecos@uclan.ac.uk

Abstract. We propose a fast subpixel motion estimation algorithm for the H.264 advanced video coding (AVC) standard. The algorithm utilizes the correlation of the spatial interpolation effect on the full-pixel motion estimation best matches between different block sizes in order to reduce the computational cost of the overall motion-estimation process. Experimental results show that the proposed algorithm significantly reduces the CPU cycles in the various motion estimation schemes by up to 16% with similar rate-distortion performance when weighed up against the H.264/AVC standard. [DOI: 10.1117/1.3095795]

Subject terms: H.264; fast motion estimation; block matching; interpolation.

Paper 080867LR received Nov. 7, 2008; revised manuscript received Dec. 23, 2008; accepted for publication Jan. 13, 2009; published online Mar. 12, 2009.

1 Introduction

The H.264 advanced video coding (AVC) standard[1] is the newest standard from the ITU-T Video Coding Experts Group and the ISO/IEC Moving Pictures Experts Group. Its main advantages are the great variety of applications in which it can be used and its versatile design. This standard has shown significant rate-distortion (RD) improvements as compared to other standards for video compression.

The standard provides great flexibility in the selection of block sizes for motion estimation/compensation, with a minimum luma block size as small as 4×4. Although most prior standards enable half-pixel motion vector accuracy at most, the H264/AVC further allows quarter-pixel motion vector accuracy for improved performance. Although the standard has shown significant RD improvements, it has also increased the overall encoding complexity due to the very refined motion-estimation (ME) process. The ME process consists of two stages: integer-pixel motion search and fractional-pixel motion search. Because the complexity of integer-pixel ME has been greatly reduced by numerous fast ME algorithms,[2,3] the computation overhead required by fractional-pixel ME has become relatively significant.

Different fast fractional-pixel ME algorithms[3–6] have been proposed, and some of them are used by the JM reference software.[7] Their common idea is to simplify the search pattern by applying very refined prediction algorithms and improved adaptive threshold schemes to terminate unnecessary search positions.

In this paper, we focus on decreasing the complexity of fractional-pixel ME by effectively applying a two-step algorithm. First, we examine the 16×16 macroblock fractional-pixel ME best match, derived from the outcome we eliminate the fractional-pixel motion search for 16×8 and 8×16 macroblock partitions. Likewise, in the second step we examine the 8×8 macroblock partitions fractional-pixel ME best matches and, derived from the outcome, we eliminate the fractional-pixel motion search for 8×4, 4×8, and 4×4 macroblock partitions.

Our algorithm differs from the previous methods in two aspects: (i) It uses the similarities between the interpolation effect on the macroblock and its partitions to completely eliminate the fractional-pixel ME. (ii) The proposed algorithm is adaptive and can be applied to any combination of integer and fractional-pixel ME schemes.

The rest of the paper is organized as follows. Section 2 gives a brief overview of the ME algorithms proposed in the H.264/AVC. Section 3 describes the proposed ME algorithm. Section 4 contains a comprehensive list of experiments and a discussion. Section 5 concludes the letter.

2 ME in the H.264/AVC

In the first stage of ME, integer-pixel motion search is performed for each square block of the slice to be encoded in order to find one (or more) displacement vector(s) within a search range. The best match is the position that minimizes the Lagrangian cost function J_{motion}

$$J_{\text{motion}} = D_{\text{motion}} + \lambda_{\text{motion}} R_{\text{motion}} \qquad (1)$$

where λ_{motion} is the Lagrangian multiplier, D_{motion} is an error measure between the candidate macroblock taken from the reference frame(s) and the current macroblock, and R_{motion} is the number of bits required to encode the difference between the motion vector(s) and its prediction from the neighboring macroblocks (differential coding). A similar functional to Eq. (1) is used to decide the optimal block size for ME. The most common error measures are the sum of absolute difference (SAD) and the sum of absolute transformed differences (SATD).

After the integer-pixel motion search finds the best match, the values at half-pixel positions around the best match are interpolated by applying a one-dimensional six-tap finite impulse response (FIR) filter horizontally and vertically. Then the values of the quarter-pixel positions are generated by averaging pixels at integer and half-pixel positions. Figure 1 illustrates the interpolated fractional pixel positions. Uppercase letters indicate pixels on the full-pixel grid, while numeric values indicate elements at half-pixel positions and lowercase letters indicate pixels in-between, at quarter-pixel positions.

For example, in Fig. 1, if the integer best match is position E, the half-pixel positions 1–8 are searched using Eq. (1). Suppose position 7 is the best match of the half pixel search. Then the quarter-pixel positions a–h are searched, again using Eq. (1).

0091-3286/2009/$25.00 © 2009 SPIE

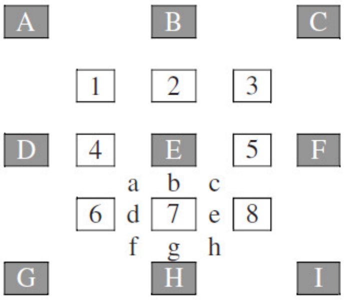


Fig. 1 Fractional pixel search positions.

Table 2 Encoder experiment conditions

Parameter	Value	Parameter	Value
Profile	100 (Main)	YUV format	YUV 4:2:0
Level IDC[7]	40	B-Frame	Not used
Entropy coding	CABAC	Frame skip	0
References	5	Search range	32
ME metric level 0	SAD	ME metric levels 1&2	Hadamard SAD

3 Proposed Scheme

In slow-motion video sequences or in the slow motion segments of fast video sequences, the ME process might find a best-match position during the integer-pixel motion search, which does not change after the subsequent fractional-pixel motion search. Furthermore, if the integer-pixel ME best match for a bigger block size does not change during fractional-pixel motion search, it is "highly likely" that this blocks' partitions integer-pixel ME result will also not change during the fractional-pixel motion search. How likely, depends on the difference in block sizes as demonstrated below. The above observations are shown in Table 1.

Table 1 is divided into three rows. The first row shows the probability that the 16×8 and 8×16 macroblock partitions have the same best match in integer- and fractional-pixel ME, given that the 16×16 macroblock has the same best match in integer- and fractional-pixel ME. We call this probability PROB(1). Similarly, the second row shows the probability of 8×8, 8×4, 4×8, and 4×4 blocks having the same best match in integer and fractional-pixel-ME, given that the 16×16 macroblock has the same best match in integer- and fractional-pixel ME. We call this probability PROB(2). The third row shows the probability of 8×4, 4×8, 8×4, and 4×4 blocks partitions having the same best match in integer- and fractional-pixel ME, given that the 8×8 blocks have the same best match in integer- and fractional-pixel ME. We call this probability PROB(3).These probabilities are averaged across sequences with different motion characteristics and are shown in the second column of Table 1.

From Table 1, it can be seen that the conditional probabilities are reasonably high ($\lesssim 70\%$) only when the macroblock/block and their partitions do not differ much in terms of size. For example, we cannot safely say that the 8×4, 4×8, and 4×4 partitions would find the same best match in the integer- and fractional-pixel motion search, given that enclosing 16×16 macroblock does so. In this case, the difference in size is big, because the 16×16 macroblock is 8, 8, and 16 times bigger with respect to the aforementioned block sizes. Using the above insights, we have developed the following scheme:

Table 1 Evaluation of the conditional probabilities.

Probabilities	Average
PROB(1)	70%
PROB(2)	59%
PROB(3)	70%

If the 16×16 macroblock finds the same best match in the integer- and fractional-pixel motion searches, then we disable the fractional-pixel motion search for all the enclosed 16×8 and 8×16 blocks. Thus, we can save all the fractional-pixel search, SAD, and Hadamard transform calculations for these blocks, Otherwise, the fractional-pixel motion search is performed.

Similarly, if the 8×8 block partitions of the 16×16 macroblock find the same best match in the integer- and fractional-pixel motion searches, we disable the fractional-pixel motion search for all the enclosed 8×4, 4×8, and 4×4 blocks. Otherwise, the fractional-pixel motion search is performed.

4 Experiments

To assess the proposed algorithm, a comprehensive set of experiments for a variety of video sequences with different motion characteristics was performed. In this experiment, the source code for the H.264 Reference Software Version JM12.2[7] was used in a Pentium-4 PC running at 2.8 GHz with 1.0 GB RAM. Table 2 illustrates the conditions of the experiments.

Table 3 shows the percentage cycle savings, the Bjontegaard Delta bit rate (BDBR) percentage differences, and the Bjontegaard Delta Peak signal-to-noise ratio (BDPSNR) differences (in decibels)[8] between the H264/AVC and the algorithm we propose when full search (FS), enhanced predictive zonal search (EPZS),[2] and unsymmetrical-cross multi-hexagon-grid search (UMHEXS)[3] are used as full and fractional-pixel ME schemes.

The Intel VTune performance analyzer was used to measure the number of machine cycles differences. Table 3 shows that the BDBR percentage differences are in the range of $[-0.5,1.2]$, while the BDPSNR differences are in the range of $[-0.04,0.02]$. The minus signs denote PSNR degradation and bit-rate savings, respectively.

Table 3 Experimental results.

		Full pixel ME		Sub pixel ME	Full pixel ME		Sub pixel ME	Full pixel ME		Sub pixel ME	Full pixel ME		Sub pixel ME	Full pixel ME		Sub pixel ME
		FFS		FS	UM HEX		UM HEX	UM HEX		FS	EPZS		EPZS	EPZS		FS
Sequence	Size	BDPSNR *(db)*	BDBR *(%)*	Cycles *(%)*	BDPSNR *(db)*	BDBR *(%)*	Cycles *(%)*	BDPSNR *(db)*	BDBR *(%)*	Cycles *(%)*	BDPSNR *(db)*	BDBR *(%)*	Cycles *(%)*	BDPSNR *(db)*	BDBR *(%)*	Cycles *(%)*
Akiyo	QCIF	+0.02	–0.45	6.5	–0.01	+0.34	14.32	0.0	0.0	14.85	–0.01	+0.24	10.9	–0.01	+0.11	14.9
	CIF	–0.04	–0.12	8.2	–0.01	+0.43	16.06	–0.02	+0.49	15.89	–0.01	+0.25	11.04	–0.02	+0.49	15.14
Foreman	QCIF	–0.05	+1.2	2.45	–0.02	+0.49	2.94	–0.02	+0.67	3	–0.03	+0.84	3.61	–0.01	+0.24	6.7
	CIF	–0.01	+0.28	2.77	–0.03	+0.72	3.17	–0.02	+0.6	3.23	–0.04	+0.81	3.23	–0.01	+0.29	3.35
Mobile	QCIF	–0.03	+0.43	1.98	–0.05	+0.51	2.09	–0.04	+0.35	2	–0.03	+0.39	2.03	–0.03	+0.28	1.3
	CIF	–0.01	+0.12	1.68	–0.02	+0.31	1.08	–0.01	+0.17	1.21	–0.01	+0.16	1.36	–0.02	+0.38	1.29
Stefan	QCIF	–0.03	+0.51	1.88	–0.03	+0.52	2.2	–0.04	+0.01	2.1	–0.13	+0.2	2.07	–0.02	+0.4	2.4
	CIF	–0.03	+0.51	1.87	–0.01	+0.12	1.7	–0,01	+0.23	2	–0.01	+0.2	1.89	–0.01	+0.24	2.2
Silent	QCIF	–0.02	+0.48	6.2	–0.01	+0.1	13.16	–0.03	+0.48	11.6	–0.02	+0.43	8.92	+0.01	–0.14	11.71
	CIF	–0.02	+0.55	6.01	–0.01	+0.37	11.79	–0.02	+0.47	12.65	–0.02	+0.46	9.37	–0.02	+0.43	12.6
Tempete	QCIF	–0.01	+0.2	1.9	–0.02	+0.25	4.5	–0.03	+0.33	1.92	–0.03	+0.37	2.14	–0.01	+0.29	1.44
	CIF	–0.01	–0.01	1.15	–0.01	+0.16	1.08	0.0	+0.06	1.4	–0.01	+0.02	1.34	–0.01	+0.18	3.05
Opening ceremony	720× 480	–0.01	+0.2	2.89	–0.01	+0.22	3.67	–0.01	+0.17	4.1	–0.01	+0.22	2.56	–0.01	+0.15	3.2
Driving	720× 480	–0.01	+0.16	1.3	–0.01	+0.12	1.45	0	+0.05	1.9	0	+0.05	1.80	–0.02	+0.08	1.1

This clearly shows that the proposed algorithm has very similar RD performance to the H.264/AVC. Furthermore, percentage cycle savings up to 16% are observed. It also can be seen that the reduction in the CPU cycles depends on the characteristics of the image sequences. For a slow image sequence with a simple background, the reduction is much more significant than for a fast image sequence or sequences with a more complex background.

5 Conclusion

In conclusion, we proposed a fast Subpixel ME based on the interpolation effect on different block sizes for H264/AVC standard. For RD performance very similar to the standard, the proposed technique can reduce up to 16% of the CPU cycles required for different ME schemes. Our scheme is very relevant to low-complexity video-coding systems.

References

1. T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.* 13(7), 560–577 (2003).
2. A. Tourapis, "Enhanced predictive zonal search for single and multiple frame motion estimation," In *Proc. of VCIP 2002*, pp.1069–1079, SPIE, Jan. 2002.
3. Z. B. Chen, P. Zhou, and Y. He, "Fast integer pel and fractional pel motion estimation for JVT," JVT-F017, 6th meeting, Awaji, Japan, Dec. 5–13 2002.
4. P. Yin, H. Y. C. Tourapis, A. M. Tourapis, and J. Boyce, "Fast mode decision and motion estimation for JVT/H.264," in *Proc. of ICIP 2003*, pp. 853–856, IEEE, Sep. 2003
5. Z. B. Chen, C. Du, J. H. Wang, and Y. He, "PPFPS—a paraboloid prediction based fractional pixel search strategy for H.26L," In *Proc. of ISCAS 2002*, pp. 9–12, IEEE, May 2002.
6. Z. B. Chen and Y. He, "Prediction based directional refinement (PDR) algorithm for fractional pixel motion search strategy," JVT-D069, 4th meeting, Klagenfurt, Austria, July 22–26, 2002
7. K. Sühring, H.264/AVC Reference Software Version JM12.2, http://iphome.hhi.de/suehring/tml/download/, Joint Video Team (2003).
8. G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," Doc. VCEG-M33, Apr. 2001.

EFFECT OF THE HADAMARD TRANSFORM ON MOTION ESTIMATION OF DIFFERENT LAYERS IN VIDEO CODING

Abdelrahman Abdelazim[a], Martin Varley[a] and Djamel Ait-Boudaoud[b]*.

[a]School of Computing, Engineering and Physical Sciences,University of Central Lancashire, Preston. PR1 2HE. UK
{AAbdelazim, MRVarley}@ uclan.ac.uk
[b]Faculty of Technology, University of Portsmouth, Portsmouth. PO1 3AH. UK
djamel.ait-boudaoud@port.ac.uk

KEY WORDS: Video Coding, Motion Estimation, Hadamard Transform, H.264

ABSTRACT:

In video coding, the most commonly used Motion Estimation distortion metrics are predominantly based on the Sum of Absolute Differences (SAD) and the Sum of Absolute Transformed Differences (SATD). Consequently the Joint Model (JM) H.264/AVC Reference Software utilises them and by default, the JM software selects the SAD as the Error Metric for Full-Pixel (first layer) motion estimation and the SATD as the Error Metric for Half and Quarter-Pixel (second and third layer respectively) motion estimation. Although SATD is much slower than SAD, it more accurately predicts quality from the standpoint of both objective and subjective metrics. In this paper, our experimental results show that the current H.264/AVC Rate-Distortion Optimisation method can have a negative impact when the SATD is applied. More specifically, although the SATD results in a lower bit rate with the same Peak Signal to Noise Ratio (PSNR) when applied in the integer pixel motion estimation with the subpel search disabled, it does not result in a better Rate-Distortion (R-D) performance when applied in the integer pixel motion estimation with the subpel search enabled, when compared to applying the SAD in the integer pixel motion estimation with the subpel search enabled.

1. INTRODUCTION

Today's hybrid video coding techniques apply motion-compensated prediction in combination with transform coding of the prediction error. This is done to reduce the bit rate of video signals. In recent video coding standards such as H.264/AVC (Wiegand, et al., 2003) there are seven different block sizes that can be used for motion-compensated prediction. Furthermore, to enhance the coding efficiency, the standard allows quarter-sample prediction signal accuracy.

Previously, the motion-compensated prediction result that provides the minimal distortion was widely accepted as the prediction signal. However, in recent years, it has been realised that such a selection is not always the most efficient, since the minimal distortion may result in a high bit rate, thereby degrading the overall coding performance. To solve this problem, the Rate-Distortion Optimisation (RDO) concept has been introduced. RDO techniques minimise the distortion under a constraint on the rate. A classical solution to the RDO problem is the Lagrangian optimisation which is used in the H264/AVC standard. The basic idea of this technique is to convert the RDO problem from a constrained problem to an unconstrained problem.

The Lagrangian cost function is divided into two parts; Distortion and Rate. The Distortion measurement quantifies the quality of the reconstructed pictures while the Rate quantifies the bits needed to code the macroblock. The Lagrange multiplier is usually calculated in a heuristic way or in an analytical way based on Rate-Distortion (R-D) models (Wiegand & Girod 2001) & (Li, et al., 2009).

The JM software allows the user to select the motion estimation distortion metric between the Sum of Absolute Differences (SAD), the Sum of the Squared Differences (SSD) and the Sum of Absolute Transformed Differences (SATD), the latter uses the Hadamard Transform. This has been employed to improve the rate-distortion performance and to facilitate the standard to gain much support in a variety of application areas. In this paper, the implications of the SATD based Motion Estimation on different layers are discussed. Moreover, a comparison between the SAD and SATD effect on the coefficients bits and motion vector bits on different layers is presented and analysed. In addition, future work to improve the R-D performance is proposed.

The paper is organised as follows. Section 2 gives a brief overview of the motion estimation proposed in the H.264/AVC. Section 3 describes the implications of the SATD based motion estimation on different layers, details and discussion of a comprehensive list of comparative experimental results. Section 4 concludes the paper.

2. MOTION ESTIMATION IN H.264/AVC

In the first stage of ME, an integer-pixel-motion-search is performed for each square block of the slice to be encoded in order to find one (or more) displacement vector(s) within a search range. The best match is the position that minimises the Lagrangian cost function J_{motion}:

$$J_{motion} = D_{motion} + \lambda_{motion} R_{motion} \qquad (1)$$

where λ_{motion} is the Lagrangian multiplier, D_{motion} is an error measure between the candidate macroblock taken from the reference frame(s) and the current macroblock and R_{motion} stands for the number of bits required to encode the difference between the motion vector(s) and its prediction from the neighbouring macroblocks (differential coding). A similar function to equation (1) is used to decide the optimal block size for motion estimation.

The most common error measures are the Sum of Absolute Difference (SAD) and the Sum of Absolute Transformed Differences (SATD). In particular, for any given block of pixels, the SAD between the current macroblock and the reference candidate macroblock is computed using the following equation:

$$SAD = \sum_{ij} | C_{ij} - R_{ij} | \qquad (2)$$

where C_{ij} is a pixel of the current macroblock and R_{ij} is a pixel of the reference candidate macroblock.

After the integer-pixel-motion-search finds the best match, the values at half-pixel positions around the best match are interpolated by applying a one-dimensional 6-tap FIR filter horizontally and vertically. Then the values of the quarter-pixel positions are generated by averaging pixels at integer and half-pixel positions. Figure 1 illustrates the interpolated fractional pixel positions. Upper-case letters indicate pixels on the full-pixel grid, while numeric pixels indicate pixels at half-pixel positions and lower case pixels indicate pixels in between at quarter-pixel positions [1] and [6].

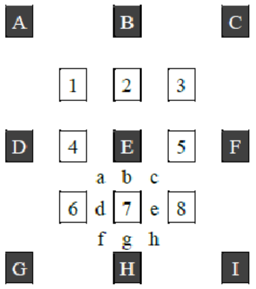


Figure 1 – *Fractional pixel search positions.*

For example, in the figure above if the integer best match is position E, the half-pixel positions 1, 2, 3, 4, 5, 6, 7, 8 are searched using equation (1). Suppose position 7 is the best match of the half pixel search. Then the quarter-pixel positions a, b, c, d, e, f, g, h are searched using again equation (1).

The Lagrangian cost can also be minimised in the frequency domain, in a very similar manner to the pixel domain. As mentioned above, SATD can be used in equation (1) instead of SAD. Central to the calculation of SATD is the 4x4 Hadamard transform which is an approximation to the 4x4 DCT transform. The transform matrix used is shown in Figure 2 below (not normalised):

$$\mathbf{H} = \begin{array}{|c|c|c|c|} \hline 1 & 1 & 1 & 1 \\ \hline 1 & 1 & -1 & -1 \\ \hline 1 & -1 & -1 & 1 \\ \hline 1 & -1 & 1 & -1 \\ \hline \end{array}$$

Figure 2 – *Hadamard Transform Matrix.*

Since **H** is a symmetric matrix, it is equal to its own transpose. By using this matrix, the (SATD) is computed using equation (3) below:

$$SATD = (\sum_{i,j} | H * (C_{ij} - R_{ij}) * H |) / 2 \qquad (3)$$

where C_{ij} and R_{ij} are the same as in equation (2) and H is the matrix in figure 2. The reader should note that the application of the Hadamard transform is optional in any resolution and can be enabled/disabled in the configuration files of the standard.

3. THE IMPLICATIONS OF THE SATD-BASED MOTION ESTIMATION ON DIFFERENT LAYERS

3.1 Use of SATD in video coding

In hybrid video coding approach following the Motion Estimation that exploits temporal statistical dependencies as described in section 2, a transform coding of the prediction residual is performed to exploit spatial statistical dependencies.

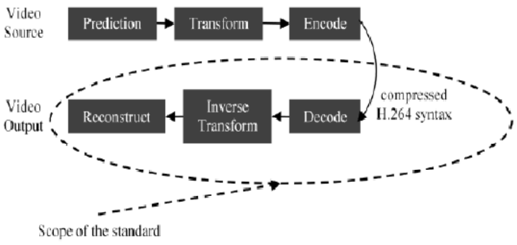


Figure 3 – *Scope of video coding standardisation*

For transform coding purposes, each colour component of the prediction residual signal is subdivided into smaller 4x4 blocks. Each block is transformed using an integer transform, and the transform coefficients are quantized and encoded using entropy coding methods.

In H.264/AVC, the transformation is applied to 4x4 blocks, and instead of a 4x4 Discrete Cosine Transform (DCT), a separable integer transform with similar properties as a 4x4 DCT is used. The transform matrix is shown in figure 4.

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}$$

Figure 4 – *Integer Transform Matrix.*

The Hadamard transform is the simplest orthogonal transform and eliminates spatial redundancies of image therefore it is usually considered as some kind of coarse approximation of DCT. This can be clearly realised when figure 2 and figure 4 are compared. As a result, ME combined with Hadamard transform is expected to find optimal difference blocks with lower redundancies, which are more suitable for subsequent DCT coding.

3.2 Effect of the Hadamard transform on motion estimation of different layers

Although the above holds true when the SATD is simply compared to other error measures metrics, in the H.264 the implementation is far more complicated; as there are two main factors that affect the overall performance. The first one is the Lagrangian cost function and its associate Lagrange multiplier and the second one is the interpolation filters that provide the half-pixel and quarter pixel search position (Wedi & Musmann 2003).

In this section, to illustrate the effects of these factors two set of experiments have been carried out. Firstly, to demonstrate the effect of the Lagrangian cost function on the SATD, we disabled the subpixel Motion search and compared the SATD performance to the SAD performance in terms of the Bjontegaard Delta Bit Rate (BDBR) percentage differences and the Bjontegaard Delta PSNR (BDPSNR) differences (in dB) (Bjontegaard, 2001), the total encoding time differences and the difference in the Distortion Weight (DW) in the Lagrangian cost function, the latter reflects the impact on the number of the required bits to encode the residual coefficients and the motion information. The result is shown in table 1. Secondly, to demonstrate the interpolation effect on the SATD, we enabled the sub-pixel motion search and performed the same comparison. The result is shown in table 2.

In this experiment six kinds of video sequences with different motion characteristics were used. The "Akiyo" sequence shows slow motion and fixed background. "Foreman" is a sequence with medium changes in motion and contains dominant luminance changes. "Tempete" is a sequence of spatial detail, fast random motion and camera zoom. "Silence" is a sequence of low spatial detail and medium changes in the motion of the arms and head of the person in the sequence. "Stefan" contains panning motion and has distinct fast changes in motion. "Mobile" contains slow panning, zooming, a complex combination of horizontal and vertical motion and high spatial colour detail.

The chosen search range was 32 pixels for the full motion estimations in the H.264 'baseline' profile. The configuration file for the encoder had the following settings: Level 40, RD optimisation ON, IPPP structure, CABAC coding, and the number of reference slices was 5.

In these experiments, the source code for the H.264 Reference Software Version JM14.2 (Sühring, 2008) was used. Two sizes QCIF (176×144) & CIF (352×288) were used in an Intel Core 2 CPU 6420 @ 2.13 GHz with 3.0 GB RAM.

For clarification purposes, the measures used in the tables are briefly explained here. The minus signs denote PSNR degradation and bitrate savings respectively. Encoding Time increase is computed as follows:

$$\Delta Time = \frac{Time_{SATD} - Time_{SAD}}{Time_{SAD}} \times 100\% \qquad (4)$$

The difference in the Distortion Weight (DW) in the Lagrangian cost function is calculated as follows:

From equation (1) DW_{SAD} and DW_{SATD} are calculated using equation (5) & (6) respectively.

$$DW_{SAD} = \frac{\sum_{The\_whole\_sequence} SAD}{\sum_{The\_whole\_sequence} SAD + \sum_{The\_whole\_sequence} \lambda_{motion} R_{motion}} \times 100\% \qquad (5)$$

$$DW_{STAD} = \frac{\sum_{The\_whole\_sequence} STAD}{\sum_{The\_whole\_sequence} STAD + \sum_{The\_whole\_sequence} \lambda_{motion} R_{motion}} \times 100\% \qquad (6)$$

Then the difference is calculated using equation (7)

$$\Delta DW = \frac{DW_{SATD} - DW_{SAD}}{DW_{SAD}} \times 100\% \qquad (7)$$

Sequence	size	BDPSNR (db)	BDBR (%)	Time (%)	DW (%)
Akiyo	QCIF	+0.1	-2.1	1053	115.9
	CIF	+0.07	-1.85	1241	144.2
Foreman	QCIF	+0.14	-3.2	710.2	60.2
	CIF	+0.16	-4.16	865.2	63
Mobile	QCIF	+0.12	-1.3	474.9	19.8
	CIF	+0.1	-1.45	623.3	29.7
Stefan	QCIF	+0.08	-1.1	524.5	23.4
	CIF	+0.07	-1.12	665.8	30.6
Silent	QCIF	+0.05	-1.1	900.9	67.1
	CIF	-0.06	-1.6	1055	75.6
Tempete	QCIF	+0.1	-1.5	543.7	28
	CIF	+0.1	-1.72	714.9	75.6
Average		+0.1	-1.85	781	61.1

Table 1– *Comparison on (BDPSNR), (BDBR), encoding time and the difference in the Distortion weight in the Lagrangian cost function between the SAD and SATD when subpixel Motion search is disabled*

	size	BDPSNR	BDBR	Time	DW
Sequence		(db)	(%)	(%)	(%)
Akiyo	QCIF	-0.01	+0.07	999	118
	CIF	+0.01	+0.1	1149	141
Foreman	QCIF	-0.05	+1.05	673.9	56.72
	CIF	-0.01	+0.12	835.4	60.53
Mobile	QCIF	-0.017	+0.2	482.2	20.3
	CIF	0.1	+1.1	661	28.7
Stefan	QCIF	+0.01	+0.01	539.8	23.3
	CIF	0	+0.01	671.3	29.8
Silent	QCIF	-0.01	+0.13	865	66.3
	CIF	-0.1	+0.31	1023	74.3
Tempete	QCIF	+0.01	-0.02	541.3	29.1
	CIF	+0.1	1.65	681	41.1
Average		+0.01	+0.39	760.1	57.4

Table 2– *Comparison on (BDPSNR), (BDBR), encoding time and the difference in the Distortion weight in the Lagrangian cost function between the SAD and SATD when subpixel Motion search is enabled.*

Table 1 shows the bitrate percentage differences (BDBR) average is -1.85 while the Delta PSNR (BDPSNR) differences average is +0.1 dB. This indicates that although the Hadamard transform outperforms the SAD, it doesn't have a significant impact on the RD performance. The reason for that can also be seen from the table where the average DW value when SATD is used is approximately 60% greater than the average DW value when SAD is used. This reduces the contribution of the second part in equation (1) and produces higher motion vector bits.

Table 2 illustrates the negative effect of the interpolation on the Hadamard transform. From the table it can be seen that when the subpixel is enabled, although when the SATD is used the average total encoding time is increased by 760%, the RD performance is degraded. Since Hadamard transform aims to match frequencies instead of pixels to get a better performance in the transform/quantisation process by reducing the coefficients bits, our observation showed that the Hadamard transform successfully reduces the coefficients bits significantly, however, in some cases the Hadamard transform does not result in finding the true motion which makes the interpolation process ineffective and affects other areas due to the prediction.

Further investigations have been carried out to examine the SATD performance against the SAD performance in subpel search. The results of these investigations indicated that the Hadamard transform outperform the SAD significantly in the subpel search because the search positions are limited to 9 positions (In full pixel the number of positions = (2*search_range+1)*(2*search_range+1)) which limit the motion vector range and increase the significance of the first part in equation (1). Furthermore, the increase in the encoding time can be tolerated.

4. CONCLUSION AND FUTURE WORK

Since ultimately the transformed coefficients are coded, better estimation of the cost can be achieved by estimating the effect of the DCT with a 4×4 Hadamard transform. Although these advantages are well known and the Hadamard transform is implemented in various parts of the ME and MD processes of the standard, for the best of our knowledge no research has been carried out to investigate the effect of the λ selection and the interpolation on the SATD. The reason for this is the extensive computations required to execute the SATD; which involves subtraction, addition, shift and absolute operations. However, if the ME is improved to accommodate the use of the SATD in the fullpixel motion search, in addition to enhancing the effect of the DCT, significant RD enhancement can be achieved in wide range of applications. Particularly, in hardware applications when applying the same distortion metric at different resolutions is essential.

To overcome some of the limitations of using SATD in the full pixel Motion Search two methods can be introduced:

1) Store a few ME vector candidates (the number can vary, subject to experiments then applying the Sum of Absolute Transformed Difference (SATD) using Hadamard transform of those candidates. This should improve the bit rate by having positive effect on the DCT and minimises the effect of the above mentioned problem.

2) Train the λ as in (Wiegand & Girod 2001), but instead of using SAD use SATD.

Further research is necessary to enhance the RD performance when SATD is used the full pixel Motion Search.

REFERENCES

Bjontegaard, G., Apr. 2001. G. Calculation of Average PSNR Differences between RD-curves. Doc. VCEG-M33.

Li, X. Oertel, N. Hutter, A. & Kaup, A.,2009. Laplace Distribution Based Lagrangian Rate Distortion Optimization for Hybrid Video Coding, *IEEE Transactions on Circuits and Systems for Video Technology*, 19(2), pp.193-205.

Sühring, K., 2007. H.264/AVC Reference Software Version JM14.2 : Joint Video Team. Available at: http://iphome.hhi.de/suehring/tml/download/ (accessed Dec. 2009).

Wedi, T. & Musmann, H.G., 2003. Motion- and Aliasing-Compensated Prediction for Hybrid Video Coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7), pp.577-586.

Wiegand, T. Sullivan, G.J. Bjøntegaard, G. & Luthra, A., 2003. Overview of the H.264/AVC Video Coding Standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7), pp.560-576.

Wiegand, T. & Girod, B., 2001 Lagrange multiplier selection in hybrid video coder control. IEEE International Conference on Image Process. (ICIP), Thessaloniki, Greece, Vol.3. pp. 542-545.

Low Complexity Hierarchical Prediction Algorithm for H.264/SVC

Abdelrahman Abdelazim, Stephen Mein and Martin Varley
School of Computing, Engineering and Physical Sciences,
University of Central Lancashire, Preston. PR1 2HE. UK
{AAbdelazim, SJMein, MRVarley}@ uclan.ac.uk

Djamel Ait-Boudaoud
Faculty of Technology,
University of Portsmouth, Portsmouth. PO1 3AH. UK
djamel.ait-boudaoud@port.ac.uk

***Abstract*— In the scalable video coding extension of the H.264/AVC standard, an exhaustive search technique is used to select the best coding mode for each macroblock. This technique achieves the highest possible coding efficiency, but it demands a higher video encoding computational complexity which constrains its use in many practical applications. This paper proposes combined fast sub-pixel motion estimation and a fast mode decision algorithm for inter-frame coding for temporal, spatial, and coarse grain signal-to-noise ratio scalability. It makes use of correlation between the macroblock and its enclosed partitions at different layers. Experimental results show that the scheme reduces the computational complexity significantly with negligible coding loss and bit-rate increases when compared to JSVM 9.15 and recently reported fast mode decision algorithms.**

I. INTRODUCTION

The objective of scalable video coding is to enable the generation of a unique bitstream that can adapt to various bit-rates, transmission channels and display capabilities. The scalability is categorised into temporal, spatial, and quality. For temporal and spatial scalabilities, subsets of the bit stream (referred to as sub-streams) represent the source content with a reduced frame rate (temporal resolution) or picture size (spatial resolution), respectively. With quality scalability, the sub-stream provides the same spatial and temporal resolutions as the complete bit stream, but with a lower signal-to-noise ratio (SNR) [1].

In order to improve coding efficiency, the SVC scheme incorporates inter-layer prediction mechanisms to complement the H.264/AVC very refined Motion Estimation (ME) and mode decision processes. However, this further increases the overall encoding complexity of the scalable coding standard.

To reduce the implementation complexity, several fast techniques have been presented recently [2-4]; most of them share the same concept of using the correlation between a macroblock (MB) and its neighbours in different layers and also that between a MB in the base layer and its corresponding position in the enhancement layer. In [2] for example, a fast mode decision algorithm for inter-frame coding for temporal, spatial and quality scalability is presented. It makes use of the mode-distribution correlation between the base layer and enhancement layers. Specifically, after the exhaustive search technique is performed at the base layer, the number of candidate modes for enhancement layers is reduced.

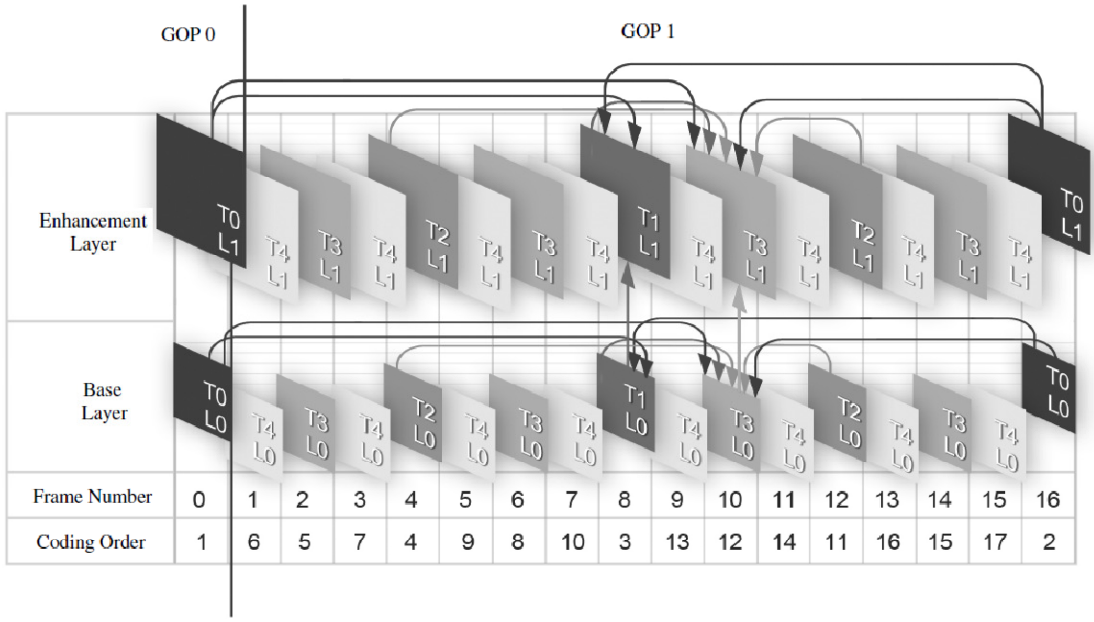
In [3] and [4] fast mode decision algorithms exploiting correlations between MBs and their neighbours are proposed. These algorithms have an obvious limitation when applied to fast video sequences or sequences with complex backgrounds, as the irregularity between the MB and its neighbours increases, which results in limited benefits and significant performance degradation.

In the movement toward the Next Generation Video Coding (NGVC) H.265 adaptive interpolation filters specifically adapted to statistics of the current image are introduced in [5], and motion compensated prediction with 1/8-pel displacement vector resolution is discussed in [6]. However, little research has been carried out to investigate and define situations when the interpolation process is unnecessary, an issue addressed in [7]. In this paper the algorithm proposed in [7] is extended to a Group of Pictures (GOP) in H.264/SVC. Moreover, the algorithm has been utilised in the mode selection process of different scalability layers to reduce the computational complexity.

The paper is organised as follows. Section 2 gives a brief overview of the hierarchical prediction proposed in the H.264/SVC. Section 3 describes the proposed mode decision algorithm, while section 4 presents details and discussion of a comprehensive list of comparative experimental results. Section 5 concludes the paper, with a summary of findings.

II. GROUP OF PICTURES HIERARCHICAL PREDICTION

In H.264/SVC a Group of Pictures (GOP) is an encoding of a contiguous subset of frames from a video sequence; each GOP consists of two successive frames at the lowest temporal base layer (the first of which is considered to belong to the previous GOP), plus all higher resolution frames between the two. All information required to decode any one frame from the GOP is contained within it. Using the GOP concept, SNR gains of more than 1 dB can be obtained for medium bit rates when compared to the widely used IBBP coding structure [1].



Frame Number	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Coding Order	1	6	5	7	4	9	8	10	3	13	12	14	11	16	15	17	2

Figure1. An example of a 2 layer GOP of size 16 for enabling temporal and spatial scalability.

Figure 1 shows an example of a 2 layer GOP of size 16. This example shows a dyadic coding structure as it is the most used structure for its demonstrated coding efficiency. Furthermore, the figure shows the base and the enhancement layers have the same frame rate, however the H.264/SVC supports non-dyadic hierarchical prediction structures and differing frame rates for the base and the enhancement layer. Frame 0 belongs to the previous GOP but is used in the prediction of the current GOP. The following subsection outlines the main principles of the hierarchical prediction when different types of scalability are used and when sub-pixel motion estimation (half-pixel and quarter-pixel motion search) is enabled.

A. *Temporal scalability*

The enhancement layer pictures are typically coded as B-pictures, where the reference picture lists 0 and 1 are restricted to the temporally preceding and succeeding picture, respectively, with a temporal layer identifier less than the temporal layer identifier of the predicted picture. In the figure, frames 0 and 16 are coded first as intra-frames, then frame 8 is coded as a B-picture using frame 0 and 16 as references. Then the rest of the frames are coded in the order shown in the figure. As an example when frame 10 is coded, up to 3 references can be used in list 0 (left references), and 2 references in list 1 (right references). Additionally bi-prediction signal, formed by a weighted sum of list 0 and list 1, can be used. In the JSVM software [8] the exact number of references is a user-configurable parameter.

B. *Spatial scalability*

In each spatial layer, motion-compensated prediction and intra-prediction are employed similar to the processes used for the base layer. Additionally an inter-layer prediction mechanism is incorporated as shown in figure 1. For instance, when frame 10 is coded, in addition to the 5 possible references described in the temporal scalability, information from the corresponding frame in the base layer can be used in motion-compensated prediction.

C. *Quality scalability*

Quality scalability is considered as a special case of spatial scalability with identical picture sizes for base and enhancement layers, therefore the same motion-compensated predictions including the inter-layer prediction are employed.

D. *Motion prediction stages*

In the first stage of motion prediction, an integer-pixel motion search is performed for each 16×16 square MB from the frame to find the best match within a given search range in the reference frames. After this, the values at half-pixel positions and quarter-pixel positions around the best match are searched to find an even better match. This process is repeated for the eight enclosed partitions of the MB (two 16×8, two 8×16, and four 8×8), and the mode that minimises a cost function is selected as the best encoding mode. Furthermore, if the four 8×8 blocks are selected as the best mode the search is repeated for all eight enclosed sub-MBs (two 8×4, two 4×8, and four 4×4). In addition, the skip mode, direct mode and intra modes are also supported.

Although the exploitation of different macroblock sizes significantly improves RD performance, this action increases the complexity of the ME process. This is due to the fact that for every macroblock partition, both integer and fractional Motion Estimation/Compensation have to be performed before deciding on the best mode.

III. PROPOSED ALGORITHM

In [7] it has been observed that there is a high correlation between the MB and its enclosed partitions when estimating the motion at different resolutions. Therefore a two step fast sub-pixel motion estimation scheme based on this observation has been developed.

1) In the first step, if the 16×16 MB finds a best match in the full-pixel motion search that does not change after performing the sub-pixel motion search (cond_1), then the sub-pixel motion search for all the enclosed 16×8 and 8×16 blocks is disabled.

1) Similarly, if in the second step the 8×8 block partitions of the 16×16 MB find the same best match in the full and sub-pixel motion searches (cond_2), the sub-pixel motion search for all the enclosed 8×4, 4×8 and 4×4 sub-blocks is disabled.

In the following sub-sections, several conditional probabilities that relate cond_1 and cond_2 to the motion estimation and the mode distribution within different layers of the H264/SVC have been evaluated. Moreover, additional conditional probabilities of the mode distribution have been calculated. Only three sequences are shown below, similar outcomes were obtained for other sequences.

TABLE 1. Conditional Probabilities of *P1* and *P2*

Sequence	QP	*P1 (%)*	*P2 (%)*			
			T1	T2	T3	T4
City	10-42	94.39	95.29	96.02	95.75	97.67
Harbour		97.02	94.06	98.97	99.13	91.73
Hall		99.28	99.53	99.68	98.54	99.57

A. *Probability of mode selection in temporal scalability*

The first conditional probabilities focus on measuring the probabilities of the mode distribution of the different temporal layers in relation to cond_1 and the mode distribution of lower temporal layers. Two probabilities were analysed;

P1, the probability of the best mode belonging to the following set {Skip = Mode 0, 16×16 = Mode 1, 16×8 = Mode 2 and 8×16 = Mode 3} when cond_1 is true.

P2, the probability of the best mode belonging to the same set (the set of modes < 4), given that the best modes for all the corresponding MBs in list 0 and list 1 at lower temporal layer in the same GOP are also < 4.

This analysis was conducted using a GOP of size 16 for the different temporal layers. It reflects the average of using 9 Quantisation Parameters (QP) (10-14-18-22-26-30-34-38-42) as shown in table 1.

From the table it can be seen that the conditional probabilities in all cases are greater than 90%. Therefore, *P1* and *P2* can be used as reliable indicators for mode predication.

B. *Correlation of mode selection in Spatial and SNR scalability*

A second set of conditional probabilities were evaluated to calculate the probabilities of the mode distribution in the enhancement layers in relation to cond_1 and the mode distribution in the base layer for spatial and SNR scalability. In the study three probabilities were analysed;

P3, the probability of the best mode in the enhancement layer being < 4 given that cond_1 is true for the corresponding MB in the base layer.

P4, the probability of the best mode in the enhancement layer being < 4 given that the best mode of the corresponding MB in the base layer is < 4.

P5, the probability of the best mode in the enhancement layer being equal to the best mode of the corresponding MB in the base layer.

TABLE 2. Conditional Probabilities of *P3, P4* and *P5*

Sequence	Spatial			SNR		
	P3 (%)	*P4* (%)	*P5* (%)	*P3* (%)	*P4* (%)	*P5* (%)
City	94.39	99.19	33.88	85.09	84.37	50.02
Harbour	97.02	98.82	26.81	80.14	80.55	42.25
Hall	99.98	99.28	13.53	98.23	99.8	19.19

Table 2 shows an average of *P3*, *P4* and *P5* when;
GOP =16, enhancement and base frame rate = 15,
base QP = 14-42 and enhancement QP =12-40.

The table shows high correlation between the MB motion characteristics in the base layer and the best mode in the enhancement layers; this is reflected by the high *P3* value. Also, the mode distribution characteristics of the enhancement layer are highly correlated to the mode distribution characteristics of the base layer and therefore may be used to predict the mode in the enhancement layer. This is shown by the high *P4* value. In [2] an observation related to *P4* has been used to limit the mode selection process. The low value of *P5* demonstrates that no assumptions can be made on the base layer and the enhancement layer having the same best mode.

C. *Motion characteristics probabilities in spatial and SNR scalability*

A final set of conditional probabilities was evaluated to calculate the probabilities of the motion characteristics in the enhancement layers in relation to cond_1 and the cond_2 in the base layer. Two probabilities were calculated;

P6, the probability that cond_1 is true for a 16×16 MB in the enhancement layer, given that cond_1 is true for the corresponding MB in the base layer.

P7, the probability of cond_2 being true for an 8×8 partition in the enhancement layer, when cond_2 is true for the corresponding block in the base layer.

Similar experimental conditions as in section 3.2 were used in the evaluation. The probabilities are shown in table 3 below.

TABLE 3: Conditional Probabilities of *P6* and *P7*

Sequence	Spatial		SNR	
	P6 (%)	*P7 (%)*	*P6 (%)*	*P7 (%)*
City	87.60	89.62	85.09	83.09
Harbour	92.86	95.28	87.54	95.15
Hall	85.80	84.86	84.78	83.09

The results show that when integer motion occurs in the base layer, similar motion should be expected in the enhancement layers; on average this is correct for 88% of cases. Thus, *P6* and *P7* can be used to adaptively enable and disable the sub-pixel motion search.

D. *Low complexity hierarchical prediction algorithm*

Based on the above observations, a fast mode decision algorithm and fast sub-pixel motion estimation for inter-frame prediction were developed as follows:

1) Temporal scalability:

a) If the 16×16 finds a best match in the full pixel ME that does not change after performing the fractional pixel ME (Integer pixel Motion) then;

i) limit the mode to 16×16, 16×8 and 8×16,
ii) disable sub-pixel ME for the 16×8 and 8×16.

b) If the collective mode in previously encoded MB in lower temporal layers is less than 4, then limit the mode selection to 16×16, 16×8 and 8×16.

c) If none of the above is true and if the 8×8 finds a best match in the full pixel ME that does not change after performing the fractional pixel ME (Integer Pixel Motion); then disable the sub pixel ME for the 8×4 , 4×8 and 4×4.

2) Spatial and SNR scalability:

For the temporal scalability in addition to *a), b)* and *c)* the following three steps are added:

d) If the 16×16 MB in the base layer finds a best match in the full pixel ME that does not change after performing the fractional pixel ME (Integer Pixel Motion) then;

i) limit the mode to 16×16, 16×8 and 8×16,
ii) disable sub-pixel ME for the 16×8 and 8×16.

e) If the best mode for the correspondent block in the base layer is less than 4, then limit the mode selection to 16×16, 16×8 and 8×16.

f) If none of the above is true and if the corresponding 8×8 partition in the base layer find a best match in the full pixel ME that does not change after performing the fractional pixel ME (Integer Pixel Motion); then disable the sub-pixel ME for the 8×4 , 4×8 and 4×4.

The algorithm is summarised using the following pseudocode

```
BEGIN

IF (MB $\in$ Spatial or SNR Base Layer)
{
   IF (cond_1)
         {
          $MODE_{CURRENT\_MB} < 4$;

          Disable sub-pixel Motion Search for the
          enclosed 16×8 and 8×16 blocks;
         }

  ELSIF (All the best modes in the previously encoded
          MBs in lower temporal layers < 4)
         {
          $MODE_{CURRENT\_MB} < 4$;
         }

  ELSIF (cond_2)
         {
          Disable sub-pixel Motion Search for the
          enclosed 8×4, 4×8 and 4×4 blocks;
         }
}

ELSIF (cond_1 || cond_1 for corresponding Base Layer
          MB)
{
          $MODE_{CURRENT\_MB} < 4$;

          Disable sub-pixel Motion Search for the enclosed
          16×8  and 8×16 blocks;
}

ELSIF (the best mode of the corresponding Base Layer
          MB < 4)
{
          $MODE_{CURRENT\_MB} < 4$;
}

ELSE IF (cond_2 OR cond_2 for correspondent Base
          Layer MB)
{
 Disable sub-pixel Motion Search for the enclosed 8×4, 4×8
and 4×4 blocks;
}

Decide best mode using the method described in section 2;

RETURN;

END
```

IV. EXPERIMENTAL RESULTS

To evaluate the proposed algorithm, a comprehensive set of experiments have been carried out. The scheme is implemented on a JSVM 9.15 encoder [8]. The test platform uses an Intel Core 2 CPU 6420 @ 2.13 GHz with 2.0 GB RAM. The Intel VTune performance analyzer was used to measure the number of machine cycles differences which reflects the total encoding Time Saving (TS).

Additionally, Bjontegaard delta PSNR (BDPSNR), and Bjontegaard delta bit rate (BDBR) [9] have been used to evaluate the proposed algorithm performance versus the JSVM encoder and recent work in the area. The test conditions are shown in Table 4.

1) Temporal scalability:

The experimental results are given in Table 5. In the table, negative values denote PSNR degradation and bit rate savings respectively. It can be seen that our scheme achieves an average of 48.5% time saving with a negligible PSNR losses and bit rate increments.

TABLE 4. Encoder Experiment Conditions

Parameter		Value	Parameter	Value
Resolution	Base	QCIF	GOP size	16
	Enhancement	CIF	MV resolution	¼ Pel
QP Setting	Base	14-22 30-38	No. of frames	100
	Enhancement	12-20 28-36	Motion Search range	16
Frame rate in/ out	Base	15HZ	Reference Picture	5
	Enhancement		Search Function	SAD

TABLE 5. Comparison between the Proposed Algorithm and the JSVM 9.15 software.

Sequence	TS (%)	BDPSNR (dB)	BDBR (%)
City	41.8	-0.09	1.89
Foreman	36.9	-0.15	2.5
Mobile	39.2	-0.16	1.5
Harbour	43.1	-0.12	1.1
Silent	56.3	-0.11	2.3
News	60.2	-0.1	2
Hall	62.5	-0.14	2.6
Average	48.5	-0.12	1.98

2) *Spatial and SNR scalability:*

In this experiment, initially our algorithm is compared to the mode decision algorithm in JSVM9.15 software; the comparison result is shown in table 6. It can be observed from this table that our algorithm provides a time reduction in the range of 35–56% and 36–62%, depending on the video content, while generating comparable video quality for scalable spatial and scalable quality video coding respectively.

Table 7 provide comparison results between a recently developed fast mode selection algorithm [3] and our method. It shows that our method achieves an average of 18.5% and 13.4% time saving for scalable spatial and scalable quality video coding respectively, while maintaining better PSNR and lower bit rate.

TABLE 6. Comparison between the Proposed Algorithm and the JSVM 9.15 software.

Sequence	Spatial			SNR		
	TS (%)	BD PSNR (dB)	BD BR (%)	TS (%)	BD PSNR (db)	BD BR (%)
City	37.2	-0.08	2.16	41.5	-0.09	1.77
Foreman	36	-0.1	2.4	36.5	-0.1	3.06
Mobile	35.4	-0.1	2	39.1	-0.	1.7
Harbour	37.7	-0.05	1.2	44.2	-0.08	1.5
Silent	48.9	-0.1	2.3	56.1	-0.14	2
News	51.5	-0.09	2	58.1	-0.12	1.8
Hall	56.6	-0.05	2.1	62.2	-0.07	2.2
Average	43.3	-0.08	2.02	48.2	-0.09	2

TABLE 7. Comparison between the Proposed Algorithm and Ren's Method [3].

Sequence	Spatial			SNR		
	TS (%)	BD PSNR (dB)	BD BR (%)	TS (%)	BD PSNR (db)	BD BR (%)
City	24.6	+0.25	-1.75	19.8	+0.08	-1.1
Foreman	7.75	+0.14	-2.27	4.4	+0.13	-1.7
Mobile	18.4	+0.07	-0.58	8.6	+0.08	-0.87
Harbour	28.7	+0.03	-0.2	26.4	+0.09	-2.32
Silent	10.7	+0.07	+0.11	13.1	+0.09	-1.3
News	9.5	+0.34	-2.09	10.2	+0.21	-2.07
Hall	30.1	+0.17	-0.6	11.7	+0.15	-0.69
Average	18.5	+0.16	-1.1	13.4	+0.11	-1.44

V. CONCLUSION

In this paper, we present a fast sub-pixel motion estimation and fast mode decision algorithm for inter-frame coding in SVC by exploiting the correlation between a macroblock and its partitions in different layers. When compared to the JSVM software this algorithm achieves a reduction of 45% encoding time on average, with a negligible average PSNR loss and bit-rate increase in temporal, spatial and SNR scalability. This saved computation can advance the progress in the realisation of the H.264 scalable extension in real-time applications and low complexity coding systems.

REFERENCES

[1] H. Schwarz, D. Marpe, and T. Wiegand, Overview of the Scalable Video Coding Extension of the H.264/AVC Standard, *IEEE Trans. Circuits Syst. Video Technology*, vol.17, no.9, pp.1103-1120, Sep. 2007.

[2] H. Li, Z. Li, and C.Wen, Fast Mode Decision Algorithm for Inter-Frame Coding in Fully Scalable Video Coding, IEEE Trans. Circuits Syst. Video Technology, vol. 16, no.7,pp.889-895, Jul. 2006.

[3] J. Ren and N. Kehtarnavaz "Fast Adaptive Early Termination for Mode Selection in H.264 Scalable Video Coding" IEEE International Conference on Image Processing (ICIP'08), San Diego, California, U.S.A. Oct. 2008.

[4] G. Goh, J. Kang, M.Cho and K. Chung "Fast Mode Decision for Scalable Video Coding based on Neighboring Macroblock Analysis" ACM Symposium on Applied Computing, pp.1845 ~ 1846, 2009.

[5] D. Rusanovskyy, K. Ugur, and J. Lainema "Adaptive Interpolation with Directional Filters" Video Coding Experts Group (VCEG) 33rd Meeting: Shenzhen, China, 20 October, 2007.

[6] J. Ostermann, and M. Narroschke "Motion compensated prediction with 1/8-pel displacement vector resolution" Video Coding Experts Group (VCEG) 30th Meeting: Hangzhou, China, 23-27 October, 2006.

[7] A. Abdelazim, M. Yang, C. Grecos, and D. Ait-Boudaoud "Selective application of sub-pixel motion estimation and Hadamard transform in H.264/AVC" Proc. SPIE, Vol. 7244, 72440C,Feb. 2009.

[8] J. Reichel, H. Schwarz, and M. Wien, J. Reichel, and M. Wien, "Joint Scalable Video Model 9.15 (JSVM 9.15)," Joint Video Team, Doc. JVT-V202, March 2009.

[9] G. Bjontegaard, Calculation of Average PSNR Difference between RD-Curves, Document VCEG-M33, Apr. 2001.

Fast Motion Prediction Algorithm for Multi-View Video Coding

Abdelrahman Abdelazim[a], Guang Y. Zhang[a], Stephen James Mein[a], Martin Roy Varley[a]
and Djamel Ait-Boudaoud[b].
[a] University of Central Lancashire, UK,
{AAbdelazim, GYZhang1 SJMein, MRVarley}@ uclan.ac.uk
[b] University of Portsmouth. UK, djamel.ait-boudaoud@port.ac.uk

ABSTRACT

Multiview Video Coding (MVC) is an extension to the H.264/MPEG-4 AVC video compression standard developed with joint efforts by MPEG/VCEG to enable efficient encoding of sequences captured simultaneously from multiple cameras using a single video stream. Therefore the design is aimed at exploiting inter-view dependencies in addition to reducing temporal redundancies. However, this further increases the overall encoding complexity

In this paper, the high correlation between a macroblock and its enclosed partitions is utilised to estimate motion homogeneity, and based on the result inter-view prediction is selectively enabled or disabled. Moreover, if the MVC is divided into three layers in terms of motion prediction; the first being the full and sub-pixel motion search, the second being the mode selection process and the third being repetition of the first and second for inter-view prediction, the proposed algorithm significantly reduces the complexity in the three layers.

To assess the proposed algorithm, a comprehensive set of experiments were conducted. The results show that the proposed algorithm significantly reduces the motion estimation time whilst maintaining similar Rate Distortion performance, when compared to both the H.264/MVC reference software and recently reported work.

Keywords: Multiview, Motion Estimation, H.264, Mode Decision, Inter-View, MVC, Sub-Pixel

1. INTRODUCTION

With the wide expansion of 3D and free viewpoint video applications, the H.264 Multiview Video Coding (MVC) standard has been developed as an extension to the H.264 Advanced Video Coding (AVC) standard to enable efficient coding for scenes captured from multiple cameras [1, 2]. Since all cameras capture the same scene from different viewpoints, inter-view statistical dependencies can be expected. Therefore, in addition to the H.264/AVC very refined Motion Estimation and mode decision processes, H.264/MVC exploits inter-view prediction for more efficient coding. This leads to a significant increase in the encoding time.

Most of the fast prediction algorithms that were designed for and applied to H.264/AVC can be implemented in any view of the MVC views. However, due to the inter-view flexibility in the MVC, the number of the possible references for any frame is far more than the AVC. Recently proposed fast MVC prediction algorithms [3], [4] and [5] reduce encoder complexity by locating corresponding objects in neighbouring views by means of a global disparity vector and exploiting the mode distribution correlation between neighbouring views. These algorithms perform well only for certain video sequences and camera configurations; given that the inherent macroblock characteristics are not taken into account.

In [6], an object-based fast prediction mode decision method has been proposed. Segmentation is used to divide the frames into foreground and background objects. First, motion-based segmentation is applied to non-anchor frames by using information of both motion vectors and intensity value. Then, the disparity-based segmentation is carried out by considering distribution of disparity vectors in the reference anchor picture. After the segmentation, Inter-view prediction is only employed for MBs in the foreground regions. The downside of this algorithm is applying several complex pre-processing steps for the segmentation purpose which limits the overall gain.

In [7] we proposed a combined fast sub-pixel motion estimation and a fast mode decision algorithm for inter-frame coding in SVC video coding. It makes use of correlation between the macroblock and its enclosed partitions at different layers. In this paper the algorithm proposed in [7] is adapted and extended to inter-view prediction in the MVC. The main difference between the proposed algorithm and other fast prediction schemes is that it exploits macroblock inner information instead of depending on neighbour and collocated macroblocks information, thus using more reliable sources especially in the case of fast video sequences.

This paper is organised as follows. The MVC concepts and requirements are outlined in section 2. The proposed algorithm is presented in Section 3. Experimental results are presented in Section 4. Finally, a conclusion is provided in Section 5.

2. OVERVIEW OF H.264/ MULTIVIEW VIDEO CODING (MVC)

The stereo high profile of the MVC standard has been standardised in June 2009. MVC stream is backward compatible with H.264/AVC, which allows older devices and software to decode stereoscopic video streams, ignoring additional information for the second view [8]. The MVC main targeted applications are Free Viewpoint Television (FTV) and Multiview 3D television. Two main challenges face most multiview applications; the first one is the transmission of huge amount of data, which requires the development of highly efficient coding schemes, and the second is that any compression scheme designed specifically for multiview video streams should support random access functionality, allowing viewers to access arbitrary views with minimum time delay. Therefore, a set of requirement has been laid out for implementation of the MVC[9]; the main three requirements were large gain compared to independent compression of each view, temporal random access and view random access.

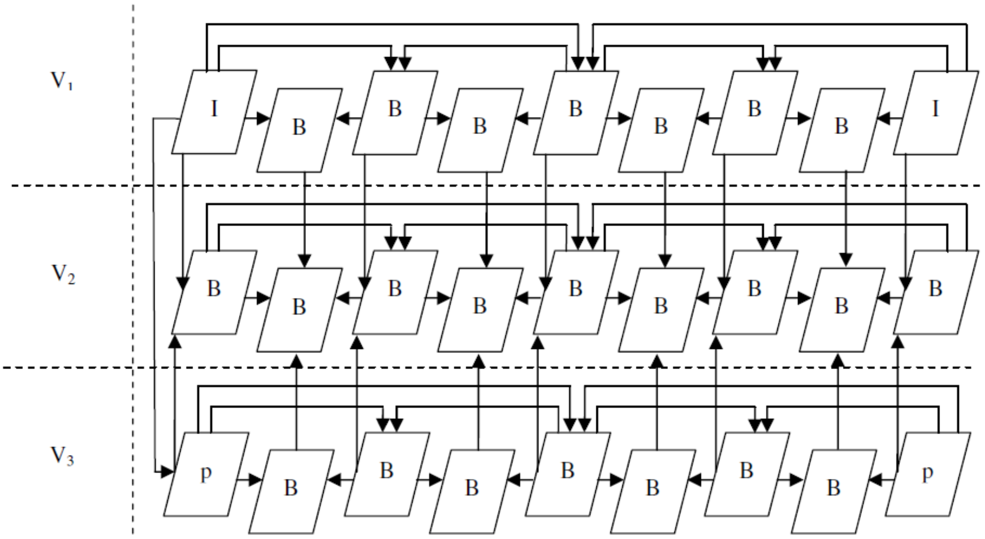
Several analyses[10] have been carried out to investigate the effectiveness of inter-view prediction. The result of these analyses indicated that for a significant number of MBs inter-view prediction is frequently found to be more efficient than temporal prediction, although for all video sequences temporal prediction is the most often chosen mode. An averages of 15% of MBs in various multiview video sequences find their best matches in neighbouring views. This can lead to a considerable bit rate reduction but at the expense of increasing the encoder complexity.

To maintain a good balance between the encoding complexity and the bit rate saving, a number of prediction structures have been proposed for the MVC project. The majority of these algorithms based on the multiple reference picture technique in the H.264/AVC. Among all the schemes for MVC, the one proposed by Heinrich Hertz Institute (HHI) in [11] has achieved the best performance, and adapted to the MVC reference software [12]. The scheme is shown in figure 1.

In the figure the hierarchical B pictures [13] are employed as this is considered the most efficient temporal prediction structure. The first picture of the first video sequence is intra-coded as an IDR picture and so-called key pictures, referred to as I picture in the figure. Then at regular intervals, defined by the Group of Pictures (GOP) size, frames are coded as I frame. For the second and third views, V_2 and V_3 respectively, all I frames are encoded as P or B pictures. This leads to significant saving in the number of encoded bits due to the fact that I frames require substantially more bits than B or P frames.

The down side of this method is that individual views can no longer be encoded or decoded independently as they share reference pictures. Furthermore, designing the encoder and the decoder becomes more complex, specifically for managing references frames and data buffers.

The MVC standard reference software [12] allows great flexibility in encoding multiviews videos. This is accomplished by employing number of user-configurable parameters in the software main configuration file. The user firstly inputs the number of views and the coding order. Then the user has the flexibility to select which view is used as references and how many references are used for key and non-key frames, given that the reference view is previously encoded.



Display Order	1	2	3	4	5	6	7	8	9
Coding Order	0	5	3	6	2	7	4	8	1

Figure 1. Inter-view prediction for key pictures and non-key pictures

3. PROPOSED ALGORITHM

This section describes the proposed novel fast prediction algorithm. In contrast to the other algorithms that were discussed in the introduction, this proposed algorithm does not depend on motion prediction results of the co-located MB in the same view or in different views. Furthermore, no thresholds are used to maintain the balance between the RD cost and reducing the complexity. Instead the MB internal information is exploited to reduce the complexity.

The proposed algorithm significantly reduces the complexity at all stages of the motion prediction in the MVC. The algorithm takes advantage of the proven fact [4, 5] that only fast moving objects in any view tend to find their best matches in neighbour views. It also takes advantage of the fast motion and mode selection algorithm that have been proposed in [7], which utilises the correlation between a MB and it enclosed partitions' motion results in different layers to define areas in any frame with integer motion, those areas may also be classed as homogeneous areas.

An additional step has been incorporated, extending the algorithm proposed in [7] to limit the use of inter-view prediction to fast moving objects, thereby reducing the overall complexity. The algorithm can be summarised in the following steps:

1) If the 16×16 finds a best match in the full pixel ME that does not change after performing the fractional pixel ME (Integer pixel Motion) then
   i. Disable inter-view prediction,
   ii. Limit the mode to 16×16, 16×8 and 8×16

iii. Disable the sub pixel ME for the 16×8 and 8×16.

2) If the 8×8 find a best match in the full pixel ME that does not change after performing the fractional pixel ME (Integer pixel motion);then disable the sub pixel ME for the 8×4 , 4×8 and 4×4.

The algorithm is shown in figure 2 as a flowchart.

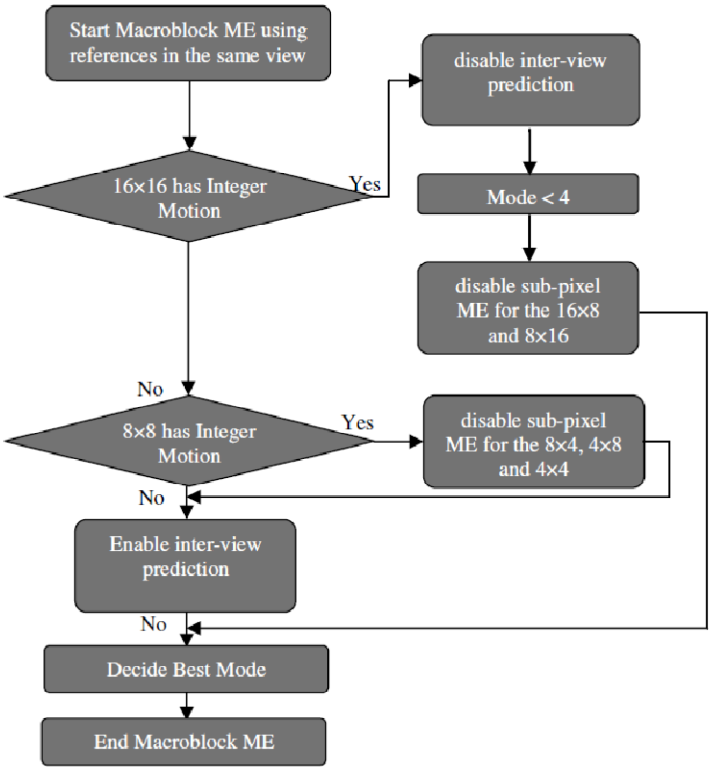

Start Macroblock ME using references in the same view
16×16 has Integer Motion
Yes
disable inter-view prediction
Mode < 4
disable sub-pixel ME for the 16x8 and 8x16
No
8x8 has Integer Motion
Yes
disable sub-pixel ME for the 8x4, 4x8 and 4x4
No
Enable inter-view prediction
No
Decide Best Mode
End Macroblock ME

Figure 2. The proposed algorithm.

: main advantage of this method in comparison to other schemes is that it makes use of some of the standard available ls to find homogeneity instead of employing additional pre-processing steps to segment the frames. Also, no litional statistical analyses need to be carried out nor its results need to be stored.

4. EXPERIMENTS

evaluate the proposed algorithm, a comprehensive set of experiments have been carried out. The scheme is plemented on a JMVM 8.0 encoder [12]. An Intel Core(TM) i7 CPU 920 @ 2.47 GHz with 8.0 GB RAM running ndows 7 was used. The Intel VTune performance analyzer was used to measure the number of machine cycles ferences, reflecting the total Time Saving (TS).

litionally, Bjontegaard delta PSNR (BDPSNR), and Bjontegaard delta bit rate (BDBR) [14] have been used to luate the proposed algorithm performance versus the JMVM encoder

Eight standard data sets[15] that vary in the number of cameras/views, the arrangement of the cameras, distance between cameras, as well as properties of the images in terms of image size and frame rate were used. All sequences were in YUV 4:2:0 planar formats. The test conditions are shown in Table 1.

Table 1. Encoder Experiment Conditions.

Parameter	Value	Parameter	Value
Resolution	640x480	GOP size	16
	1024x768	MV resolution	¼ Pel
QP Setting	14-22 30-38	No. of frames	200-300
		Motion Search range	32
Frame rate in/ out	15-25 and 30 HZ	Reference Picture	2
		Search Function	SAD

The experiment results are shown in table 2. For clarification purposes, the measures used in the tables are briefly explained here. The negative values denote PSNR degradation and bitrate savings respectively. Time saving TS is computed as follows:

$$TS = \frac{T_{jmvm} - T_{proposed}}{T_{jmvm}} \times 100\,\% \qquad (1)$$

Table 2. Comparison between the Proposed Algorithm and the JMVM 8.0 software.

Sequence	TS (%)	BDPSNR (dB)	BDBR (%)
Akko&kayo	55.4	-0.09	1.09
Flamenco	53.6	-0.06	1.62
Race	42.8	-0.03	0.86
Rena	66.5	-0.1	1.04
Uli	52.3	-0.01	0.73
Ballet	76.12	-0.02	0.92
Breakdancing	63.55	-0.1	0.93
Exit	64.8	-0.05	1.34
Average	59.41	-0.05	1.06

It can be seen that the proposed scheme achieves an average of 59.38% time saving with negligible losses in PSNR and increments in bit rate. Additionally the proposed algorithm has been compared with the recently proposed algorithm [6]. The comparison is shown in table 3.

Table 3. Comparison between the Proposed Algorithm and the Algorithm proposed in [6].

Sequence	TS (%)	BDPSNR (dB)	BDBR (%)
Akko&kayo	29.4	-0.01	0.15
Flamenco	27.8	-0.04	0.21
Race	19.6	0.06	0.27
Rena	25.6	-0.03	0.12
Uli	24.2	-0.02	1.01
Ballet	20.2	-0.01	-0.01
Breakdancing	18.5	0.05	-0.73
Exit	21.4	-0.11	-0.143
Average	23.33	-0.014	0.10

From the above tables it can be seen that for a very similar RD performance the proposed algorithm achieves an average of 23.33% time saving when compared to [6]. Although the time saving is dependent upon video content, the proposed scheme results in significant time savings when compared to H.264/MVC reference software and other reported work.

5. CONCLUSION

The large amount of video data and particularly high computational complexity make the MVC encoder difficult to be applied in real time applications. This paper presents a fast algorithm for multi-view video coding. The experimental results have shown that more than half of time consumed in the prediction process is saved with only negligible loss of performance.

The proposed algorithm depends on the property of video sequences that fast moving objects are likely to be predicted using inter-view references while background object is more likely to be predicted using references from the same view. The MBs inherited correlation is exploited as a motion-based segmentation to locate fast moving objects.

The resulting multiview prediction structure has the advantage of achieving significant coding gains whilst being highly flexible in regard to its adaptation for various of spatial and temporal setups.

REFERENCES

[1] ISO/IEC/JTC1/SC29/WG11," Multi-view Coding using AVC", Bangkok, Thailand, Jan. 2006.

[2] Ho, Y., S., and Oh, K., J., "Overview multi-view video coding" Proc. Systems, Signals and Image Processing, 5-12 (2007).

[3] Li, X., Zhao, D., Ji, X., Wang, Q., and Gao, W., "A Fast Inter Frame Prediction Algorithm For Multi-View Video Coding" Proc. ICIP, 417-421(2007).

[4] Shen, L., Yan, T., Liu, Z., Zhang, Z., An, P., and Yang, L. "Fast Mode Decision For Multiview Video Coding" Proc. ICIP, Cairo, Egypt, pp. 2953-56,Sept. 2010.

[5] Shen, L., Yan, T., Liu, Z., Zhang, Z., and An, P.,, "Selective Disparity Estimation and Variable Size Motion Estimation Based on Motion Homogeneity for Multi-View Coding", IEEE Trans. on Broadcasting Video Technol., 55(4),761-766(2009).

[6] Lee, S., Y., Shin, K., M., and Shin, K., M., "An Object-based Mode Decision Algorithm for Multi-view Video Coding" Proc. ISM, 74-81(2008).

[7] Abdelazim, A., Mein, S., Varley, M., Grecos, C. and Ait-Boudaoud, D "Fast multilayered prediction algorithm for group of pictures in H.264/SVC" Proc. SPIE 7871, (2011).

[8] Smolic, A., "Introduction to Multiview Video Coding", ISO/IEC JTC 1/SC 29/WG 11N9580 , (2008).

[9] ISO/IEC JTC1/SC29/WG11, "Requirements on Multiview Video Coding v.4.", Doc. N7282, (2005).

[10] Merkle, P., Smolic, A., Mueller, K., Wiegand, T., 2007. Efficient prediction structures for multiview video coding" IEEE Transon Circuits Syst. Video Technol., 17(11):1461-1473,(2007)

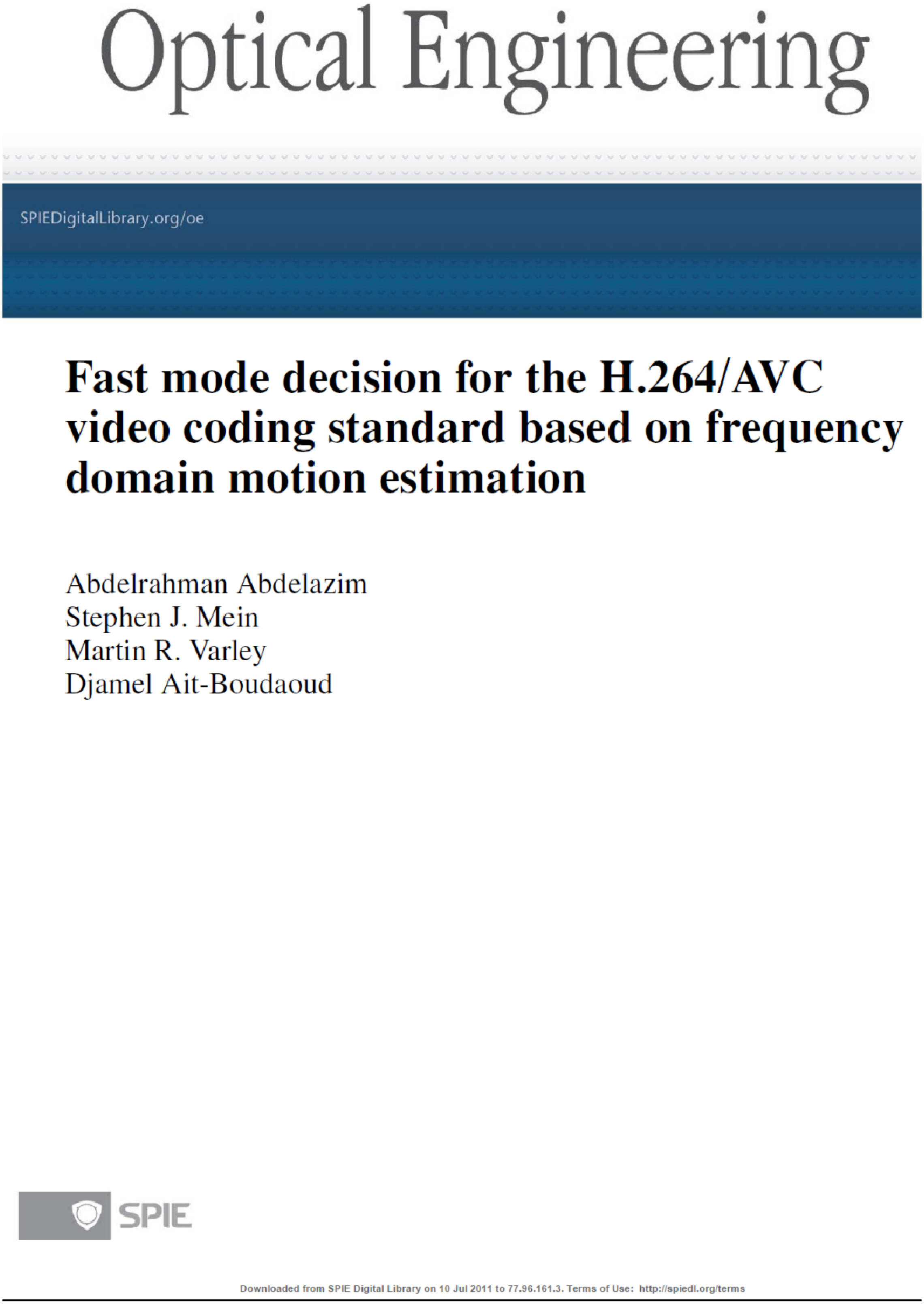
[11] JTC1/SC29/WG11, MPEG2006/W7798, "Description of Core Experiments in MVC", ISO/IEC (2006)

[12] Vetro, A., Pandit, P., Kimata, H., and Smolic, A., "Joint Multiview Video Model (JMVM) 8.0," ISO/IEC JTC1/SC29/WG11 and ITU-T Q6/SG16, Doc. JVT-AA207, (2008).

[13] Schwarz, H., Marpe, D., and Wiegand, T., "Analysis of hierarchical B pictures and MCTF", proceeding IEEE International Conference on Multimedia and Expo, (2006).

[14] Bjontegaard, G., "Calculation of Average PSNR Difference between RD-Curves", Document VCEG-M33, (2001).

[15] Su, Y., Vetro, A., and Smolic, A., "Common Test Conditions for Multiview Video Coding" ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6), Document: JVT-T207, , 15–21 (2006)



Optical Engineering

SPIEDigitalLibrary.org/oe

Fast mode decision for the H.264/AVC video coding standard based on frequency domain motion estimation

Abdelrahman Abdelazim
Stephen J. Mein
Martin R. Varley
Djamel Ait-Boudaoud

Fast mode decision for the H.264/AVC video coding standard based on frequency domain motion estimation

Abdelrahman Abdelazim, Stephen J. Mein, Martin R. Varley, and Djamel Ait-Boudaoud
University of Central Lancashire, Preston, Lancashire, PR1–2HE United Kingdom
E-mail: AAbdelazim@uclan.ac.uk

Abstract. The H.264 video coding standard achieves high performance compression and image quality at the expense of increased encoding complexity. Consequently, several fast mode decision and motion estimation techniques have been developed to reduce the computational cost. These approaches successfully reduce the computational time by reducing the image quality and/or increasing the bitrate. In this paper we propose a novel fast mode decision and motion estimation technique. The algorithm utilizes preprocessing frequency domain motion estimation in order to accurately predict the best mode and the search range. Experimental results show that the proposed algorithm significantly reduces the motion estimation time by up to 97%, while maintaining similar rate distortion performance when compared to the Joint Model software. *© 2011 Society of Photo-Optical Instrumentation Engineers (SPIE).* [DOI: 10.1117/1.3597609]

Subject terms: motion estimation; phase correlation; frequency domain; mode decision; H.264 advanced video coding.

Paper 101102LPRR received Dec. 28, 2010; revised manuscript received May 4, 2011; accepted for publication May 17, 2011; published online Jul. 6, 2011.

1 Introduction

The H.264 advanced video coding (AVC) standard[1] is the newest standard from the ITU-T video coding experts group and the ISO/IEC moving pictures experts group. Its main advantages are the great variety of applications in which it can be used and its versatile design. This standard has shown significant rate distortion (RD) improvements, as compared to previous standards for video compression.

Although the standard has shown significant RD improvements, it has also increased the overall encoding complexity due to the very refined motion estimation (ME) and mode decision processes where variable block size ME is employed. In H.264, there are seven different block sizes that can be used in intermode coding (16×16 = mode 1, 16×8 = mode 2, 8×16 = mode 3, 8×8 = mode 4, 8×4 = mode 5, 4×8 = mode 6, and 4×4 = mode 7). In addition, the SKIP mode (mode 0), direct mode, and two intramodes (INTRA_4 and INTRA_16) are also supported. To achieve the highest coding efficiency, the encoder tries all the possible modes and selects the best one which minimizes the RD cost.

However, this method is not computationally efficient, and consequently limits the use of H.264 encoders in real-time applications. Therefore, algorithms which can reduce computational complexity of H.264 encoding without compromising coding efficiency are very desirable for real-time implementation of H.264 encoders.

Several fast mode decisions[2–6] have been proposed in the literature. This section provides a review of some existing fast intermode decision techniques and their limitations.

In Ref. 2 the mean of absolute difference between the current and the co-located block in the reference frames have been used to predict the modes. This scheme achieved up to 48% computational cost reduction. A similar algorithm has been proposed in Ref. 3, where the sum of the absolute difference value of the current MB is calculated and compared to a threshold. Based on the comparison results, the modes are selected adaptively.

In Ref. 4, the motion vector information has been used to predict the modes and the scheme utilizes the spatial property of the motion vector to predict the modes efficiently.

Another fast intermode decision algorithm based on temporal correlation of modes in *P* slices was proposed in Ref. 5. A time reduction of 57% on average was claimed, with a bitrate increment of 0.07% and a loss of 0.05 dB, as compared to the standard. However, if the local temporal information is unreliable, for example, when the scene changes, the RD performance will be degraded because of mode misprediction.

A recently developed algorithm was proposed in Ref. 6. This scheme achieves up to 63% time savings when compared to the standard reference software. However, the algorithm is based on heuristic analysis obtained from a set of video sequences which can lead to a significant RD degradation if the algorithm is used to encode sequences with different characteristics. Furthermore, the spatial correlations between MBs have been exploited and this correlation is unreliable for sequences with a complex background.

From the information above, it can be seen that fast intermode decision algorithms can achieve time savings in the range of 40% to 65% with some RD performance degradations. It also can be noticed that all the fast intermode decision schemes are based on spatial domain ME information.

Recently, there has been a lot of interest in motion estimation techniques operating in the frequency domain. These are commonly based on the principle of cyclic correlation and offer well-documented advantages in terms of computational efficiency due to the employment of fast algorithms. One of the best-known methods in this class is phase correlation,[7] which has become one of the ME methods of choice for a wide range of professional studio and broadcasting applications.[8] In addition to computational efficiency, phase correlation offers key advantages in terms of its strong response to edges and salient picture features, its immunity to illumination changes and moving shadows, and its ability to measure large displacements. Several attempts[9,10] have been proposed to adapt the phase correlation to the standard. In Ref. 9, the authors proposed an adaptive block size phase correlation ME, which has been compared to the full search block matching (FSBM) algorithm.[11] The comparison results indicated a significant increase in the bitrate. Furthermore, block sizes up to 32×32 were used to estimate the motion which increases the computational complexity. In Ref. 10, the authors used the phase correlation to predict the ME block size by generating a binary matrix, and then selected the

0091-3286/2011/$25.00 © 2011 SPIE

mode from the binary matrix. Although the authors claimed a 50% reduction in the ME time, the algorithm showed significant RD performance degradation for slow video sequences.

In this paper, we propose a novel fast mode decision algorithm. In addition to saving up to 97% of the ME time for similar RD performances, our algorithm differs from the above-mentioned algorithms as it preprocesses the macroblock in the frequency domain using 16×16 phase correlation, and based on these results, we directly predict the mode and the search range.

The rest of the paper is organized as follows. Section 2 describes the proposed mode decision algorithm. Section 3 contains a comprehensive list of experiments and a discussion. Section 4 concludes the paper.

2 Proposed Scheme

In video compression, knowledge of motion helps to exploit similarity between adjacent and nearby frames in the sequence, and remove the temporal redundancy between neighboring frames in addition to the spatial and spectral redundancies.[12] The phase correlation method measures the movement between the two fields directly from their phases. The basic principles are described below.

Assuming a translational shift between the two frames:

$$s_t(x, y) = s_{t+1}(x + \Delta x, y + \Delta y). \tag{1}$$

Their two-dimensional (2D) Fourier transforms are:

$$S_t(f_1, f_2) = S_{t+1}(f_1, f_2) \exp[2j\pi(f_1 \Delta x + f_2 \Delta y)]. \tag{2}$$

Therefore, the shift in the spatial-domain is reflected as a phase change in the spectral domain. The cross-correlation between the two frames is:

$$C_{t,t+1}(f_1, f_2) = S_{t+1}(f_1, f_2) \cdot S_t(f_1, f_2). \tag{3}$$

The normalized cross-power spectrum is:

$$R_{t,t+1}(f_1, f_2) = \frac{S_{t+1}(f_1, f_2) \cdot S_t^*(f_1, f_2)}{|S_{t+1}(f_1, f_2) \cdot S_t^*(f_1, f_2)|}. \tag{4}$$

From Eqs. (2) and (4), we have:

$$R_{t,t+1}(f_1, f_2) = \exp[-2j\pi(f_1 \Delta x + f_2 \Delta y)]. \tag{5}$$

The 2D inverse transform is given by:

$$c_{t,t+1}(x_1, y_1) = \delta(x_1 - \Delta x, y_1 - \Delta y). \tag{6}$$

The displacement can be found by using the location of the pulse in Eq. (6). The maximum correlation is achieved when the two images are identical [value = 1 at (0, 0)]. Our observation on the phase correlation results for different images extracted from different video sequences revealed that if the correlation between the macroblock and its prediction is greater than or equal to 0.8; 92% of the time the macroblock contains objects that have a minimum size of 16×8 or 8×16 and the motion vector has a maximum value of 8 in any direction. On the other hand, when the correlation is less than 0.8, this indicates that the contents of the macroblock are either large objects with large movements or a number of small objects with various movements.

Using the above insights, we developed the following algorithm: if the correlation value is equal to 1, then we choose the SKIP mode as the best mode. Otherwise, if the correlation value is greater than or equal to 0.8, we limit the mode selection process to modes {0, 1, 2, and 3}. Additionally, we limit the search range to 8. Finally, if the correlation value is less than 0.8, we enable all the modes and ME is performed using the defined search range. The proposed algorithm is shown in flowchart form in Fig. 1.

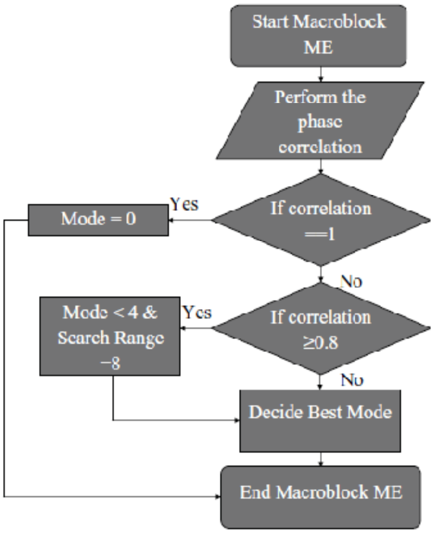


Fig. 1 The proposed algorithm.

3 Experimental Results

To assess the proposed algorithm, a comprehensive set of experiments for eight kinds of video sequences with different motion characteristics was performed.

The chosen search range was 32 pixels for the full ME. The configuration file for the encoder had the following settings: RD optimization ON, IPPP structure, CABAC coding, and the number of reference slices was 1.

In these experiments, the source code for the H.264 Reference Software Version JM14.2 (Ref. 11) was used. Four sizes, QCIF (176×144), CIF (352×288), (640×480), and (1024×768) were used in an Intel Core 2 CPU 6420 @ 2.13 GHz with 2.0 GB RAM. The Intel VTune performance analyzer was used to measure the number of machine cycles differences, reflecting the total encoding time.

Table 1 shows the percentage cycle savings, the percentage search point savings, the Bjontegaard Delta bit rate (BDBR) percentage differences, and the Bjontegaard delta peak signal-to-noise ratio (BDPSNR) differences (in decibels)[13] between the JM software and the proposed new algorithm, and between the proposed algorithm and the algorithm proposed in Ref. 13. In the first comparison, Table 1 shows that the BDBR differences are in the range of 0.2 to 1.3, while the BDPSNR differences are in the range of −0.08 to −0.01. The minus signs denote PSNR degradation and bitrate savings, respectively. This clearly shows that the proposed algorithm has very similar RD performance to H.264/AVC reference software. Furthermore, ME time savings up to 97% and percentage cycle savings up to 67% are

Table 1 Comparison on BDPSNR and BDBR cycle differences and ME time saving between the proposed algorithm and JM software and the algorithm proposed in Ref. 3.

		Against the JM software				Against the algorithm proposed in Ref. 3		
Sequence	Size	BDPSNR (dB)	BDBR (%)	Cycles Saving (%)	ME Time Saving (%)	BDPSNR (dB)	BDBR (%)	ME Time Saving (%)
Akiyo	QCIF	−0.08	+1.3	66.98	97.02	0.03	−0.4	48.12
	CIF	0.04	+0.96	57.36	86.49	0.04	0.24	42.65
Foreman	QCIF	−0.05	+1.22	36.43	45.17	0.06	−0.7	22.53
	CIF	−0.04	+1.14	35.74	44.68	0.02	−0.05	21.45
Tempete	QCIF	−0.01	+0.61	39.75	44.31	0.04	0.03	24.09
	CIF	−0.05	+0.76	40.07	46.8	0.01	−0.45	26.76
Silent	QCIF	−0.01	+0.36	60.53	80.9	0.03	0.51	50.32
	CIF	−0.03	+0.77	52.69	75.02	0.06	0.25	47.22
Stefan	QCIF	−0.03	+0.3	30.54	36.53	0.05	0.61	19.66
	CIF	−0.04	+0.5	29.39	35.81	0.06	−0.32	18.55
Mobile	QCIF	−0.02	+0.2	26.06	34.88	0.07	0.01	22.49
	CIF	−0.05	+0.6	27.4	32.74	0.01	0.83	20.87
Rena	640×480	−0.05	+0.8	42.5	64.5	−0.03	0.9	29.8
Uli	1024×768	−0.03	+0.6	39.6	51.8	0.05	−0.04	26.72
Average		−0.04	+0.7	41.8	55.4	0.04	0.1	30.08

observed. It also can be seen that the reduction in the CPU cycles depends on the characteristics of the image sequences. For a slow image sequence with a simple background, the reduction is much more significant than for fast image sequences or sequences with a more complex background. The reason for this is that in slow video sequences, the number of big block sizes increases significantly.

The second comparison in Table 1 indicates that the proposed algorithm consistently outperforms a recently proposed approach[3] in all aspects; an average of 30% encoding time savings, 0.04 dB PSNR improvement, and 0.1% total bit rate reduction.

Moreover, when comparing the results to the results in Ref. 10, in addition to the significant time reduction gain (40%), the RD performance is maintained similar to the JM software for the various sequences, while in Ref. 10, the performances have been degraded rather significantly for some of the sequences.

4 Conclusion

The H.264/AVC increases memory bandwidth and spends a significant amount of processing time for the motion estimation process in order to determine the optimal motion vector. As a means of increasing the coding efficiency, in this paper, we proposed a fast mode decision and a motion estimation scheme with rate distortion performance similar to the standard. Our technique can reduce up to 97% of the ME time (67% in CPU cycles), resulting in significant time/cycle savings as compared to H.264/AVC. It is very relevant to low complexity video coding systems.

References

1. T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.* **13**(7), 560–576 (2003).
2. X. Jing and L. P. Chau, "Fast approach for H.264 inter-mode decision," *Electron. Lett.* **40**(17), 1050–1052 (2004).
3. J. Bu, S. Lou, Ch. Chen, and J. Zhu, "A predictive block-size mode selection for inter frame in H.264," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol 2, pp. 917–920, IEEE, Toulouse, France (2006).
4. L. Shen, Z. Liu, Z. Zhang, and X. Shi, "Fast inter mode decision using spatial property of motion field," *IEEE Trans. Multimedia* **10**(6), 1208–1214 (2008).
5. B. G. Kim, "Novel inter-mode decision algorithm based on macroblock tracking for the p-slice in H.264/AVC video coding," *IEEE Trans. Circuits Syst. Video Technol.* **18**(2), 273–279 (2008).
6. H. Zeng, C. Cai, and K. K. Ma, "Fast mode decision for H.264/AVC based on macroblock motion activity," *IEEE Trans. Circuits Syst. Video Technol.*, **19**(4), 491–499 (2009).
7. J. J. Pearson, D. C. Hines, S. Goldman, and C. D. Kuglin, "Video rate image correlation processor," *Proc. SPIE* **119**, 197–205 (1977).
8. G. A. Thomas, "Television motion measurement for DATV and other applications," *BBC Res. Dept. Rep.* (1987).
9. Y. Ismail, M. Shaaban, and M. Bayoumi, "An adaptive block size phase correlation motion estimation using adaptive early search termination technique," *IEEE International Symposium on Circuits and Systems*, pp. 3423–3426, IEEE, New Orleans, LA (2007).
10. M. Paul and G. Sorwar, "An efficient video coding using phase-matched error from phase correlation information," *IEEE 10th Workshop on Multimedia Signal Processing*, pp. 378–382, IEEE, Cairns, Australia (2008).
11. Source code link: http://iphome.hhi.de/suehring/tml/download/old_jm/jm14.2.zip.
12. C. Stiller and J. Konrad, "Estimating motion in image sequences," *IEEE Signal Processing Magazine*, 15(4), 70–91 (1999).
13. G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," *Doc. VCEG-M33* wftp3.itu.int/av-arch/video-site/0104_Aus/VCEG-M33.doc (2001).