COMPLEXITY ADAPTATION IN VIDEO ENCODERS FOR POWER LIMITED PLATFORMS

by

Chanyul Kim

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF DOCTOR PHILOSOPHY

in the

School of Electronic Engineering

Dublin City University

Supervisor : Prof. Noel E. O'Connor

January 2010

Declaration of Authorship

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy, is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: _____

ID number: _____

Date: _____

"Yesterday is history. Tomorrow is a mystery. Today is a gift, that's why it is called the present "

Eleanor Roosevelt

Abstract

With the emergence of video services on power limited platforms, it is necessary to consider both performance-centric and constraint-centric signal processing techniques. Traditionally, video applications have a bandwidth or computational resources constraint or both. The recent H.264/AVC video compression standard offers significantly improved efficiency and flexibility compared to previous standards, which leads to less emphasis on bandwidth. However, its high computational complexity is a problem for codecs running on power limited platforms. Therefore, a technique that integrates both complexity and bandwidth issues in a single framework should be considered.

In this thesis we investigate complexity adaptation of a video coder which focuses on managing computational complexity and provides significant complexity savings when applied to recent standards. It consists of three sub functions specially designed for reducing complexity and a framework for using these sub functions; Variable Block Size (VBS) partitioning, fast motion estimation, skip macroblock detection, and complexity adaptation framework.

Firstly, the VBS partitioning algorithm based on the Walsh Hadamard Transform (WHT) is presented. The key idea is to segment regions of an image as edges or flat regions based on the fact that prediction errors are mainly affected by edges. Secondly, a fast motion estimation algorithm called Fast Walsh Boundary Search (FWBS) is presented on the VBS partitioned images. Its results outperform other commonly used fast algorithms. Thirdly, a skip macroblock detection algorithm is proposed for use prior to motion estimation by estimating the Discrete Cosine Transform (DCT) coefficients after quantisation. A new orthogonal transform called the S-transform is presented for predicting Integer DCT coefficients from Walsh Hadamard Transform coefficients. Complexity saving is achieved by deciding which macroblocks need to be processed and which can be skipped without processing. Simulation results show that the proposed algorithm achieves significant complexity savings with a negligible loss in rate-distortion performance. Finally, a complexity adaptation framework which combines all three techniques mentioned above is proposed for maximizing the perceptual quality of coded video on a complexity constrained platform.

Acknowledgements

I would like to take this opportunity to record my sincere thanks to all who helped me to successfully complete this research.

Special thanks to my advisor Prof. Noel.E.O'Connor and Prof. Alan F. Smeaton for the invaluable guidance, encouragement, and support during my studies. My deepest thanks are also extended to examiner Prof. Fernando Pereira at IST, Lisbon, Portugal and Dr. Gabriel-Miro Muntean for the valuable opinions on my thesis.

I also thank to Dr. Hyowon Lee, Dr. Saman Cooray, Dr. Kealan McCusker, Mr. Radha Ramachandruni, Mr.Milan Redzic for all the happy memories.

I would like to thank Samsung Electronics.Ltd, especially, vice president Dr. Yunje Oh, Dr. Ko junho, Dr. Kim Byungik, Mr. Kim youngduck, Mr. Kim sangho, Mr.Jo kyuhyung, Mr.Kim hyunsu.

I take this opportunity to thank Mr.Jangyun and Mr.Sungsichul of the Korean association of North Dublin.

I owe much of the success of this work to my wife, Sumi and daughter Seunghyun for constantly encouraging and supporting me in every possible way through the years.

Publications

- **Chanyul Kim**, Kinane A and Noel. E. O'Connor, "Reducing Complexity and Memory Accesses in Motion Compensation Interpolation in Video Codecs", Proceedings of the China-Ireland International Conference on Information and Communications Technologies (CIICT 2007), Vol. 1, No. , Dublin, Ireland, 28-29 August 2007. (pp223-230)

- **Chanyul Kim** and Noel. E. O'Connor, "Using the Discrete Hadamard Transform to Detect Moving Objects in Surveillance Video", 8th IASTED International Conference on Visualization, Imaging, and Image Processing (VIIP 2008), Palma De Mallorca, Spain, 1-3 September 2008.

- **Chanyul Kim** and Noel. E. O'Connor, "Low Complexity Intra Video Coding Using Transform Domain Prediction", International Conference on Multimedia Modeling (MMM), Lecture Notes in Computer Science, vol.5371, pp.96-107, Jan 2009.

- **Chanyul Kim** and Noel. E. O'Connor, "Low Complexity Video Compression Using Moving Edge Detection Based on DCT Coefficients", International Conference on Computer Vision Theory and Applications (VISAPP), Lisbon, Portugal, 8-9 Febrary 2009.

- **Chanyul Kim** and Noel. E. O'Connor, "Fast Intra Prediction in the Transform Domain", 19th Data Compression Conference (DCC 2009), Snowbird, UT, 16-18 March 2009.

- **Chanyul Kim** and Noel. E. O'Connor, "Complexity Adaptation in H.264/AVC Video Coder for Static Cameras", 27th Picture Coding Symposium (PCS 2009), Chicago, IL, 6-8 May 2009.

- Philip Kelly, Ciar'an 'Oconaire, **Chanyul Kim** and Noel E. O'Connor, "Automatic Camera Selection for Activity Monitoring in a Multi-camera System for Tennis", Third ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC 2009), Como, Italy, 30 Aug - 2 Sep, 2009.

- **Chanyul Kim** and Noel E. O'Connor, "Low Computational Complexity Variable Block Size (VBS) Partitioning for Motion Estimation using the Walsh Hadamard Transform", IEEE International Symposium on Multimedia (ISM 2009), San Diego, California, USA, 14-16 Dec 2009.

- **Chanyul Kim** and Noel E. O'Connor "Variable Block Size Motion Estimation (VBS-ME) using Walsh Hadamard Transform (WHT)", IEEE Transaction on Image Processing (Submitted).

- **Chanyul Kim** and Noel E. O'Connor "Sikp macroblock (MB) Detection in H.264/AVC using the Walsh Hadamard Transform (WHT)", IEEE Transactions on Circuits and Systems for Video Technology (Submitted).

Contents

List of Figures

List of Tables

Acronyms

2D-LOG	2-D Logarithmic Search.
BDBR	Bjontegarrd's Delta Bit-rate.
BDPSNR	Bjontegarrd's Delta Peak Signal to Noise.
BMA	Block Matching Algorithm.
C-D	Complexity-Distortion.
C-R	Complexity-Rate.
C-R-D	Complexity-Rate-Distortion.
CABAC	Context-Adaptive Binary Arithmetic Coding.
CAVLC	Content-Adaptive Variable Length Coding.
CCD	charged coupling device.
CDS	Conjugate Direction Search.
CNN	Cellular Nonlinear Network.
CSN	Camera Sensor Network.
DCT	Discrete Cosine Transform.
DFT	Discrete Fourier Transform.
DS	Diamond Search.
DT	Discrete Transform.
DTT	Discrete Tchebichef Transform.
ED	Euclidean distance.
EPZS	Enhanced Predictive Zonal Search.

FAR	false acceptance rate.
FFT	fast Fourier transform.
FIR	Finite Impulse Response.
FS	Full Search.
FWBS	Fast sequency ordered Walsh hadamard transform Bounding Search.
FWBSR	Fast sequency ordered Walsh hadamard transform Bounding Search for Reusable block.
FWS	Fast Walsh Search.
H.264/AVC	MPEG-4 PART 10 Advanced Video Coding.
HD	High-Definition.
HT	Hotelling Transform.
HVS	Human Visual System.
ICT	Integer Cosine Transform.
IDCT	Inverse Discrete Cosine Transform.
IDTT	Integer Discrete Tchebichef Transform.
KLT	Karhunen Lòeve Transform.
MA	Moiré Artifact.
MB	Macroblock.
MBM	Multiresolution Block Matching.
MC	motion compensation.
MD	Mode Decision.
ME	Motion Estimation.
MPEG	Moving Picture Experts Group.
MSE	Mean Squared Error.
NCC	Normalized Cross Correlation.
NTSS	New Three Step Search.
NZMB	Non Zero Macro Block.
OBM	Overlapped Block Matching.

OSA	Orthogonal Direction Search.
P-R-D	Power-Rate-Distortion.
PAD	Partial Absolute Distance.
PCA	Principal Components Analysis.
PCM	Pulse Code Modulation.
PDA	Personal Digital Assistant.
PR	Precision rate.
PSAD	Pseudo Sum of Absolute Difference.
PSM	Percentage for Skip Macro-Block.
PSNR	Peak Signal-to-Noise Ratio.
QP	Quantization Parameter.
R-D	Rate-Distortion.
RDO	Rate Distortion Optimization.
RGB	Red-Green-Blue.
RLC	Run-Length Coding.
RMSE	Root Mean Squared Error.
RSSD	Root Sum of Squared Differences.
SAD	Sum of Absolute Difference.
SATD	Sum of Absolute Transform Difference.
SSD	Sum of Squared Differences.
SSE	Streaming SIMD Extensions.
SSE	Sum of Squared Error.
SSIM	Structural SIMilarity.
SWHT	Sequency ordered Walsh Hadamard Transform.
TSS	Three Step Search.
VBS	Variable Block Size.
VLC	Variable Length Coding.
VOD	Video-on-Demand.

WHM	Walsh Hadamard Matrix.
WHT	Walsh Hadamard Transform.
ZMD	Zero Motion Detection.
ZQDCT	Zero Quantized DCT coefficients detection.

I dedicate this work to the lovers of my life, ***Sumi*** *and* ***Seunghyun***

— To know is nothing at all, to imagine is everything !

Anatole France

1 Introduction

1.1 Problem Statement

To realize video services on power limited platforms such as Personal Digital Assistants (PDAs), Camera Sensor Networks (CSNs), and mobile phones, it is necessary to leverage both performance-centric and constraint-centric signal processing techniques. The problem of resource-constrained video compression on power limited platforms has been the focus of much research for the last decades. The resource constraints can be classified as follows;

1. *Bandwidth constraints*: In traditional video applications, such as digital TV broadcasting and Video-on-Demand (VOD), content can be compressed, stored on a server, and transmitted. In this case, the major constraint is in the form of bandwidth or storage space, which determines the output bit-rate of the encoder. Therefore, the ultimate goal in this situation is to optimize the video quality under the bandwidth constraints. Rate-Distortion (R-D) optimization has been developed to model the relationship between the coding bit-rate and signal distortion. Various R-D models have been proposed to deal with the trade-off between bandwidth and performance in recent decades [19, 25, 92, 103, 115].

2. *Computational resource constraints*: Mobile video applications, such as CSNs and mobile TV, typically need to operate with limited energy. A primary factor in determining the utility or operational lifetime of the mobile

devices is how efficiently it manages its energy consumption, which is sometimes identified as managing computational complexity. These kinds of applications are always a trade-off between Complexity (C) and video quality or distortion (D). Many algorithms have been reported in the literature to reduce encoding computational complexity [13, 65, 100, 111, 112, 129].

3. *Joint computational complexity-bandwidth constraints*: In wireless video applications on mobile devices, video encoding and transmission are the two dominant power consuming operations. From a power consumption perspective, video encoding presents an inherent dichotomy. First, efficient video compression significantly reduces the amount of the video data to be transmitted, which saves a significant amount of energy. Second, more efficient video compression requires higher computational complexity and thus high power consumption is needed in processing. Ideally, we could use an analytic framework to find the best trade-off. However, it is difficult to find the theoretical optimum point since Complexity (C) and R-D performance are concepts in totally unrelated area. However, in [100], Complexity-Distortion (C-D) and R-D predict asymptotically the same results assuming that the signal has stationary and ergodic properties. Therefore, research has been performed on Complexity-Rate-Distortion (C-R-D) models to satisfy joint computational complexity-bandwidth constraints [34, 42, 110].

All the approaches cited in the above handle either the optimization of video quality of an encoder or the negotiation with bit-rate based on the estimated complexity. Most common approaches are searching a R-D and a C-R-D convex hull to find the best encoder parameters. Therefore, algorithms have been proposed for non real-time applications such as two pass coding [111]. However, the required complexity is very changeable depending on the video sequences and user preferences of the encoder. For example, when a user installs a camera sensor to detect people with a wireless function for generating alarms, power constraints and bit-rate are more critical characteristics than distortion. On the contrary, if a user wants to check who is approaching his/her property, the critical characteristics may be reversed. Also more computational complexity is required for complex scenes than static scenes, but in order to check scene characteristics, additional computational complexity is inevitable.

What contributive factors in video encoding are related to computational complexity? To provide an answer, we need to analyze the encoding complexity. Let us consider the inter coding function of an encoder. The total complexity of the encoder in traditional video compression is given by [65]

$$C_{sum} \stackrel{\text{def}}{=} C_{ME} + C_{TQ} + C_{EC} + C_{M} \tag{1.1}$$

where C_{Sum} denotes the total complexity, C_{ME} is the computational complexity of Motion Estimation (ME), C_{TQ} is the complexity of transform coding and quantisation, C_{EC} is the complexity of entropy coding including run-length coding, C_{M} is the overhead complexity not controlled by the encoder such as memory accesses and bit parsers. The complexity of ME of Equation (1.1) can be denoted as

$$C_{ME} = \sum_{i \in A} N^{i}_{ME} C^{i}_{SAD} + C_{SP} \tag{1.2}$$

where C^{i}_{SAD} represents the complexity of the Sum of Absolute Difference (SAD) in block size i. For example, the elements of A are

$$= \begin{cases} \{\{16\times16\},\{16\times8\}\}, & \text{(MPEG2)} \\ \{\{16\times16\},\{8\times8\}\}, & \text{(MPEG4)} \\ \{\{16\times16\},\{16\times8\},\{8\times16\},\{8\times8\},\{8\times4\},\{4\times8\},\{4\times4\}\}, & \text{(H.264/AVC)} \end{cases}$$

Note that SAD is one of the cost functions used to measure the distortion between two images. This can be substituted for more accurate cost functions such as Sum of Absolute Transform Difference (SATD) and Sum of Squared Differences (SSD) (see Section 6.3). However, they introduce more computational complexity than SAD. N^{i}_{ME} is the number of partitioned blocks for a specific block size i, and C_{SP} denotes the complexity of sub pixel motion compensation (MC) up to quarter-pel proposed in recent standards video coding [97, 123]. C_{TQ} is the computational complexity of DCT, IDCT, quantisation, and dequantization. C_{TQ} is only determined by the complexity of coding of Non Zero Macro Blocks (NZMBs) if only the NZMBs can be detected before processing of transform and quantisation:

$$C_{TQ} = N_{NZMB} \times C_{NZMB} \tag{1.3}$$

where N_{TQ} and C_{NZMB} represent the number of NZMBs and computational complexity of the coding operation respectively. The relationship between the complexity of entropy coding and bit rate R is denoted as in Equation (1.4), and

its computational complexity is a constant at a given bit rate.

$$C_{EC}{=}R \times C_{Bit} \tag{1.4}$$

Equation (1.1)-Equation (1.4) show that the encoding complexity at a given bit rate is affected by the number of SAD operations and the number of NZMB of each video frame. The number of total blocks can be defined as

$$N_T = N_{SKIP} + N_{NZMB} \tag{1.5}$$

where N_{SKIP} represents the number of skip Macroblock (MB) in a frame. The total controllable computational complexity at a given bit rate is depicted in Equation (1.1), which is rewritten as

$$C_{sum} = \sum_{i \in A} (N_T - N_{SKIP})^i C_{SAD}^i + (N_T - N_{SKIP}) C_{NZMB}. \tag{1.6}$$

The key issues for computational complexity adaptation in a video coder are how to cost effectively reduce the number of SAD operations and detect skip MBs. Moreover, variable partitioned block sizes are introduced in the recent video coding standards (increasing i in Equation (1.6)), which means that the computational complexity of an encoder increases significantly compared to previous standards even though new standards guarantee better R-D performance. However skip MBs are dominant as the Quantization Parameter (QP) increases (as shown in Figure 5.6). Thus there is much room for saving computational complexity at high QP.

As a result, complexity can be controlled by early skip of MBs and by reducing SAD operations to detect motion vectors. However, these algorithms introduce computational complexity since additional procedures are needed to classify blocks in advance. Therefore, *low complexity algorithms for detecting skip blocks and motion vectors* are new requirements for realizing the complexity adaptation in a video encoder.

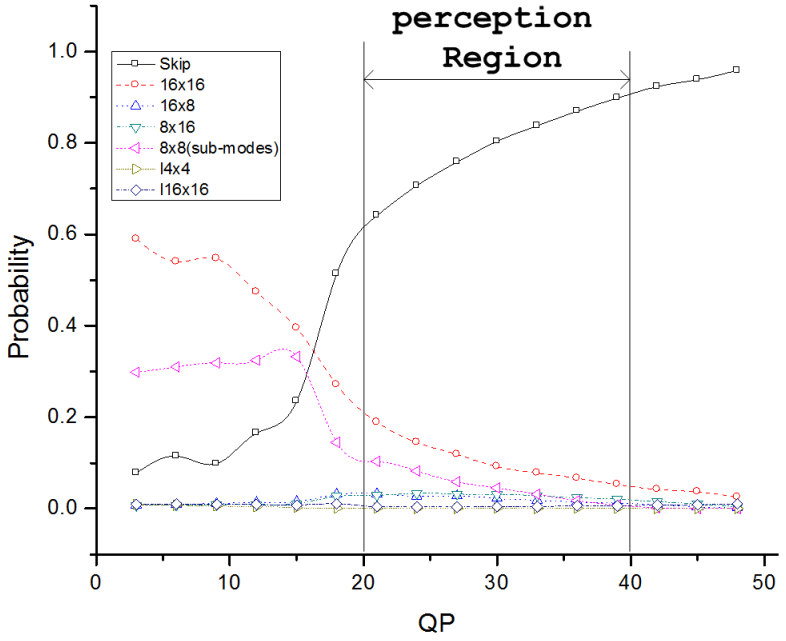


Figure 1.1: The probability of skip block of the News sequence with CIF (352×288) resolution in H.264/AVC; JM reference software (Baseline profile, 30fps)

1.2 Research Objective and Contributions

The demand for high quality and low complexity video compression increases with emerging new applications such as mobile TV and mega-pixel network cameras. The aim of this research is to present algorithms to adapt the complexity of an encoder especially targeting power limited platforms. These algorithms enable the encoder to make use of available processing resources to maximize C-D performance. The target video codec standard of this research is MPEG-4 PART 10 Advanced Video Coding (H.264/AVC) because it gives the best RD performance among existing video standards [84]. But it also requires high complexity processing and this is one of main challenges of this research. This research can be partitioned into two parts.

1. The developement of low complexity video coding sub function blocks acts on replacing complexity demanding functions such as motion estimation and DCT.

2. Complexity Adaption algorithm using proposed sub function blocks is presented as a framework.

The contributions of this research consists of four major parts depicted in Figure 1.2.

1. Low complexity pre-processing: In this procedure, the gradient features are detected by proposed low complexity algorithms. These algorithms use the Walsh Hadamard Transform (WHT) that is a simple integer transform. Furthermore these features are used for a block partitioning algorithm. This procedure works with motion estimation to be achieved low complexity sub functions of the encoder. The detail is explained in Chapter 5.

2. Motion Estimation: The most time consuming function block of an encoder is ME. It is impossible to achieve a complexity adapted video coder without low complexity algorithms for ME. This research presents a ME algorithm performed in Sequency ordered Walsh Hadamard Transform (SWHT) domain. This algorithm has computational cost effectiveness and no local minimum. Details are provided in Chapter 6.

3. Skip block detection: As QP increases, the number of skip blocks increases as well. It introduces complexity cost saving if skip blocks can be detected prior to ME, transform, quantisation, and entropy coding. This research shows that our SWHT based skip block detection algorithm is comparable to the state of the art in skip block detection algorithms. Chapter 7 introduces the skip block detection algorithm.

4. Complexity adaptation framework : Rate Distortion Optimization (RDO) introduces significant complexity cost to detect motion vectors and mode selection, and is often not feasible for real-time applications especially working on a power limited platforms. C-D is a key contribution of this research. This research focuses on managing complexity by controlling SAD and the number of skip blocks as mentioned in Section 1.1. Details are presented in Chapter 8.

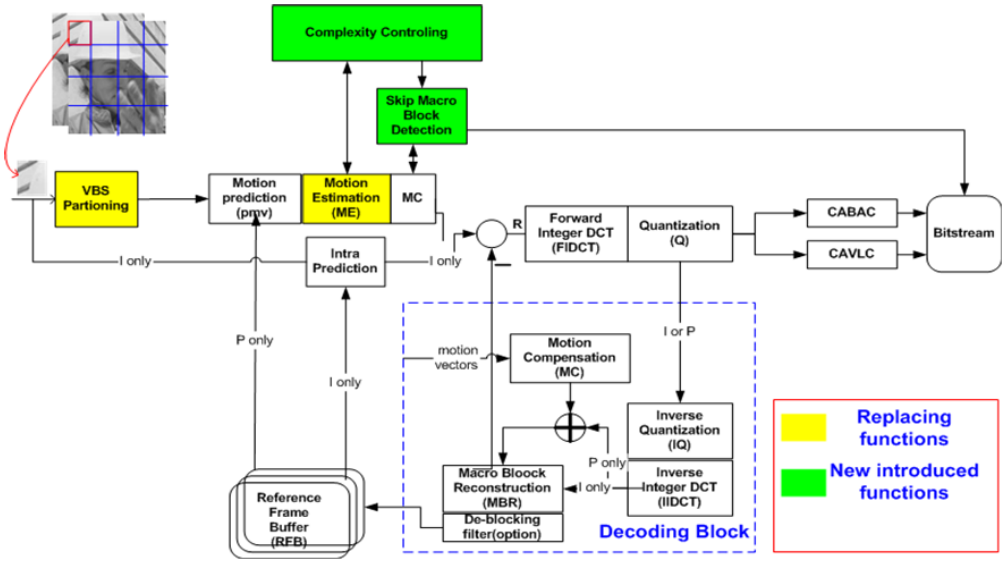


Figure 1.2: The scope of this research among all the functionalities of an encoder

1.3 Organization

The thesis is organized as follows:

- **Chapter 2** - This chapter provides some essential background knowledge on video compression. The key concepts and fundamental terms used in video compression are introduced. The main functions of a typical block based video compression are briefly explained.

- **Chapter 3** - Provides an overview of the experimental methods for a complexity adaptation video encoder. The characteristics of the test sequences and objective quality measure metrics used are explained. Moreover, the overall structure of the proposed framework is explained to provide content for Chapter 5-Chapter 8.

- **Chapter 4** - An overview of the H.264/AVC video compression standard in terms of computational complexity is provided. The contributive factors for high computational complexity, flexibility and performance are explained. This chapter explains why a large amount of computational resources are needed to implement a H.264/AVC encoder.

- **Chapter 5** - This chapter describes the Variable Block Size (VBS) partitioning algorithm based on motion edge detection. This method incorporates with fast ME algorithms to obtain further computational complexity savings.
- **Chapter 6** - The fast ME algorithm called Fast sequency ordered Walsh hadamard transform Bounding Search (FWBS) based on the WHT is proposed. Moreover, the proposed method is compared with other fast ME algorithms in terms of both performance and computational complexity.
- **Chapter 7** - This chapter presents the performance of a MB skip prediction algorithm where the relationship between the Integer Cosine Transform (ICT) and the SWHT is used. This part of the work play a important role in the complexity adaptation framework described in Chapter 8.
- **Chapter 8** - Describes a framework for complexity adaption in a H.264/AVC encoder. This new approach uses both a complexity model and the complexity control algorithm.
- **Chapter 9** - This final chapter contains the discussion and conclusion. A summary of the algorithms and a critical review of the main results are presented. Ideas for further investigation are also presented.
- **Appendix A** - Contains a mathematical derivation of the relationship between a block and its sub-blocks' SWHT coefficients. This is used for the FWBS as a low complexity tool.

1.4 Summary

The main factors related to computational complexity in a video encoder are the number of SAD operations, which could be replaced by other distortion measurement tools, and the number of skip blocks. This thesis has a target with both achieving complexity and adapting it in video encoding. Computational cost effective ME algorithms aim at reducing the number of SAD operations. On the contrary, C-D and skip block detection algorithms focus on controlling the number of skip blocks. Low complexity pre-processing based on the WHT is introduced as a basic tool underpinning the thesis contributions, which are a low

complex ME, a skip block detection, and a C-D model. Note that all proposed algorithms are performed in the WHT domain.

—By doubting we come at truth.

Marcus Tullius Cicero

2 Digital Video Coding Principals

2.1 Introduction

Video coding (sometimes called compression) refers to a process in which the amount of data used to represent video is reduced to meet a bit rate requirement, while the quality of the reconstructed video satisfies specific requirements for an application. The required quality of the reconstructed video is application dependent; for example, we may need the reconstructed video exactly the same as the original in medical diagnoses or scientific analysis, in which case the process is named lossless video coding. However, other applications such as broadcasting, video storage or transmission allow a certain amount of information loss, corresponding to lossy video coding. Lossy video coding is the main focus of this chapter. Video coding involves several fundamental concepts including encoded bit rates, visual quality of video and computational complexity. This chapter is concerned with briefly reviewing fundamental concepts of video coding.

2.2 Digital Video Representation

The representation of digital video from analog signal through sampling and digitization is depicted in Figure 2.1(a). Necessary components or functions in lossy coding are shown in Figure 2.1(b) In order to be processed by computers, an analog video that is captured by a light sensor (charged coupling device (CCD)) must be digitized. Digital video representation consists of three steps: spatial

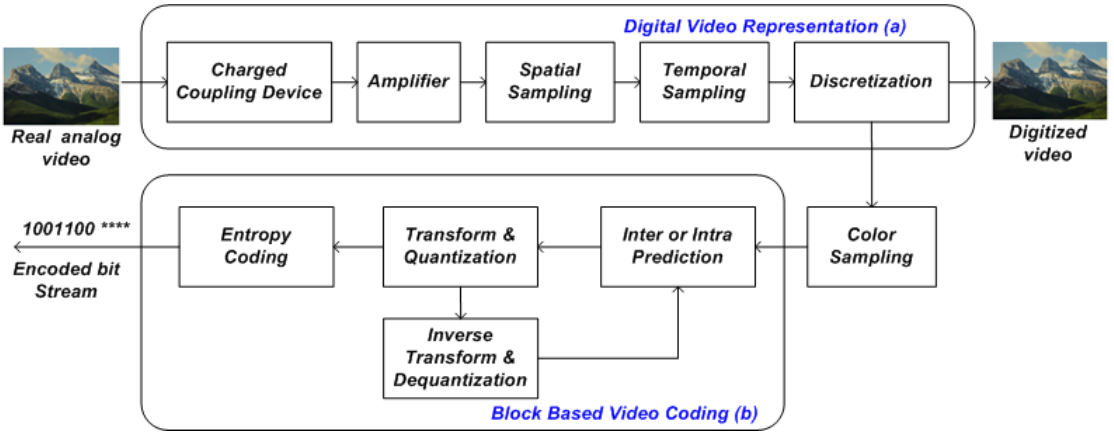


Figure 2.1: Block diagram of the digital video representation (a) and block based video coding (b)

sampling, temporal sampling, and discretization. After obtaining digitized video sequences, colour sub-sampling is an optional procedure to reduce redundancy, which is possible because the Human Visual System (HVS) is less sensitive to colour than luminance information [114].

2.2.1 Spatial Sampling

Spatial sampling consists of taking measurements of the underlying analog signal at a finite set of sampling points in a frame. The two dimensional pixel data at sampling points are transformed into a one dimensional set through raster scanning. The two main methods to perform raster scanning are progressive and interlaced as shown in Figure 2.2. In an interlaced scan, the points are divided into odd and even scan lines which make up a field so that two fields make up a frame. However, interlaced scan has some drawbacks over progressive scan such as:

- Freeze frames: Motion artifacts are accrued when an image is taken from a moment of action, to produce a freeze frame. This is caused by both fields (odd and even field) being captured at slightly different times. The first drawn field is earlier than the second drawn field in time.
- Display problem on progressive monitors: When an interlaced image is displayed on a progressive device such as a computer monitor, both fields

are displayed at once, which results in a jagged image. Therefore, a de-interlace method is needed to obtain a clear looking image on a computer monitor.

- Flickering on fine spatial detail called Moiré Artifact (MA): This is often seen in fine detail like a grille. However, these artifacts are greatly diminished and not visible in High-Definition (HD) interlaced video.

In progressive scanning, the sampling points are scanned one at a time from left to right, then moving from one row to the next, from top to bottom. Progressive scanning has been used in modern digital formats such as computer monitors, film, and so on. Note that analog television systems (NTSC, PAL) commonly use interlaced scanning, so require interlaced-progressive conversion (called de-interlacing) to display progressive digital format.

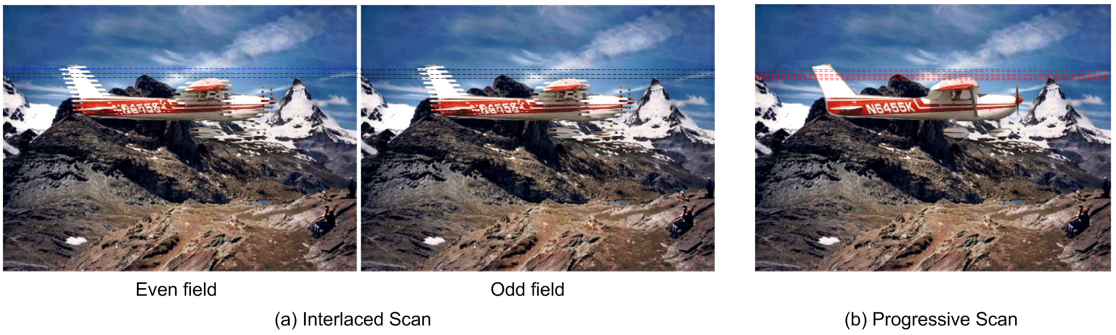


Figure 2.2: Raster scanning (a) Interlaced scanning (b) Progressive scanning

2.2.2 Temporal Sampling

The HVS is relatively slow in responding to temporal changes. There is no evidence that the HVS works in the same way as moving media with distinct frames sampled at discrete points in time. Therefore, it is difficult to express the limitations of human perception as a given maximum frame rate [68]. However, it may be possible to investigate the consequences of changes in frame rate for human observers. Based on observations, at least 16 samples per second at each grid point maintain an illusion of motion, which is the basis for motion pictures. For a film, temporal sampling is performed at a rate of 24 frame/sec. On the contrary, the sampling rate used for a television is 25 (PAL) or 30 (NTSC) frame/sec. Therefore, a conversion method should be applied to display films

on television called "Telecine" such as 3:2 pulldown for NTSC, which means 24 frame/sec is converted to 29.97 frame and 2:2 pulldown for PAL.

2.2.3 Discretization

After spatial and temporal sampling, the video signal consists of a sequence of continuous intensity values. The continuous intensity values are incompatible with digital possessing. Therefore, one more step called discretization is needed to generate a discrete set of values. It is well know that SNR increases according to increasing the number of sampling bits. Let the number of sampling bits be defined as b, then the SNR can be calculated using a uniform distribution of sampling step size as follows:

$$SNR = 10\log_{10}(2^b) \approx 6 \times b(dB). \tag{2.1}$$

This means that SNR increases by 6dB whenever the number of sampling bits increases by 1. For example, a 10-bit discretization system shows 12dB SNR enhancement compared to an 8-bit system. This process is often referred to as Pulse Code Modulation (PCM). After discretization, $N \times M$ data points called pixels or pels are obtained.

2.2.4 Colour Sampling

It is well known that the HVS is much more sensitive to luminance components than to chrominance components [78, 113]. Mithell *et al.* [74] proposed a quantitative illustration of the above statement. If luminance components is separated from chrominance components, psycho visual redundancy can be also removed from the original while keeping acceptable quality of an image. Luminance components are decoupled with chrominance components in the YUV and YCbCr colour models. The required bits for representing chrominance can be reduced by colour subsampling.

Colour model

The purpose of a colour model is to facilitate the specification of colours in some standard generally accepted way. A colour model is a specification of a 3-D coordinate system. Each industry uses the most suitable colour model for its usage. For example, the RGB colour model is used in computer graphics, YUV for analog PAL TV systems, YIQ for analog NTSC TV systems, YCbCr are mainly used in digital video systems, and so on. YCbCr has been commonly used in digital video applications such as image/video compression and other computer vision applications.

- **RGB colour model**: The Red-Green-Blue (RGB) colour system is the best known of several color systems. This is due to the following feature of human perception for colour. The colour sensitive area in the HVS consists of three different sets of cones. Each set is sensitive to the light of one of the three primary colors: red, green, and blue. Consequently, any color sensed by the HVS can be considered as a particular linear combination of the three primary colors. Moreover, the captured image from a CCD, referred to in Figure 2.1(a), has analogous data as represented by a RGB model. However, RGB is not very efficient when dealing with real images. To generate any colour within the RGB color cube, all three RGB components need to be of equal pixel depth and display resolution. Not most modification of the image requires all three colour planes. Therefore, other colour models which provide decoupling luminance with chrominance have been commonly used in image applications.

- **YUV colour model**: The YUV colour model is the basic colour model used in analogue colour TV broadcasting. Originally YUV was a re-coding of RGB for transmission efficiency (minimizing bandwidth) and for backward compatibility with black and white television. The YUV colour space is derived from the RGB space. It comprises the luminance (Y) and two colour difference (U,V) components. The luminance can be computed as a weighted sum of red, green and blue components. The colour difference or chrominance components are formed by subtracting luminance from blue and red. The principal advantage of the YUV model in image processing is decoupling of luminance and colour components. The importance

of decoupling is that the luminance component of an image can be processed without affecting its colour components. For example, the histogram equalization of the colour image in YUV format may be performed simply by applying histogram equalization to a Y component. YUV has a linear transition relationship with gamma-corrected RGB components as follow.

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{2.2}$$

- **YCbCr colour model**: The YCbCr colour space is used for digital video and was developed as part of the ITU-R BT.601 Recommendation. It should be noted that U and V may be negative in the YUV model. Therefore, it cannot be directly used in digital images. In order to make chrominance components nonnegative, the Y, U, and V are scaled and shifted to produce the YCbCr model. This model is widely used in the JPEG and MPEG-series international coding standards. The conversion matrix between gamma-corrected RGB and YCbCr is denoted as

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix}. \tag{2.3}$$

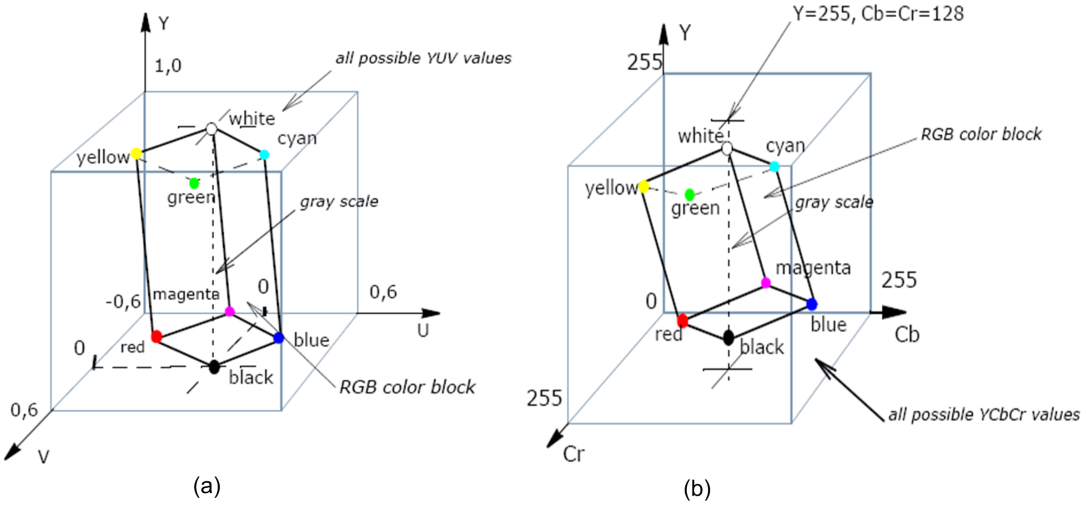


Figure 2.3: RGB colour cube [2] (a) in YUV (b) in YCbCr

Figure 2.3 represents the relationship between the RGB and YUV, YCbCr models. It shows that not all the possible values in YUV or YCbCr represent possible

RGB colours. Therefore, special care must be taken regarding overflow or underflow in RGB, when converted from YCbCr.

Colour subsampling

Figure 2.4 shows different colour subsampling methods and associated memory size. These colour subsampling methods are classified as follows;

- **4:4:4 YCbCr**: This is a format with no subsampling of Y, Cb and Cr components, which are sampled at every pixel. If RGB pixels are used for 4:4:4 subsampling, no psycho visual redundancy is achieved.
- **4:2:2 YCbCr**: This format uses 2:1 horizontal down sampling. This means that the Y component is sampled at each pixel, while Cb and Cr components are sampled at every two pixels in the horizontal direction. Therefore, the total storage for Cb and Cr is reduced by 50%.
- **4:1:1 YCbCr**: This uses 4:1 horizontal down sampling. This means that the Y component is sampled at each pixel, while Cb and Cr components are sampled every 4 pixels horizontally. The total storage of Cb and Cr requires only 25% compared to the 4:4:4 YCbCr format.
- **4:2:0 YCbCr**: This uses 2:1 horizontal down sampling and 2:1 vertical down sampling. Y is sampled at each pixel, Cb and Cr are sampled at every block of 2×2 pixels. Total storage of Cb and Cr is the same as 4:1:1 YCbCr because only the sampling direction is changed. This format has been widely used in video/image coding applications. 4:2:0 YCbCr has been used as the main format tested in this thesis.

2.3 Block Based Video Coding

Traditional block based hybrid video coding has been widely used and adapted by the international video coding standards. The idea is that a whole frame is divided into blocks pre-defined size called macroblocks (16×16 blocks are used for Moving Picture Experts Group (MPEG) standards), and blocks are encoded individually using prediction, transform and quantisation, and entropy coding.

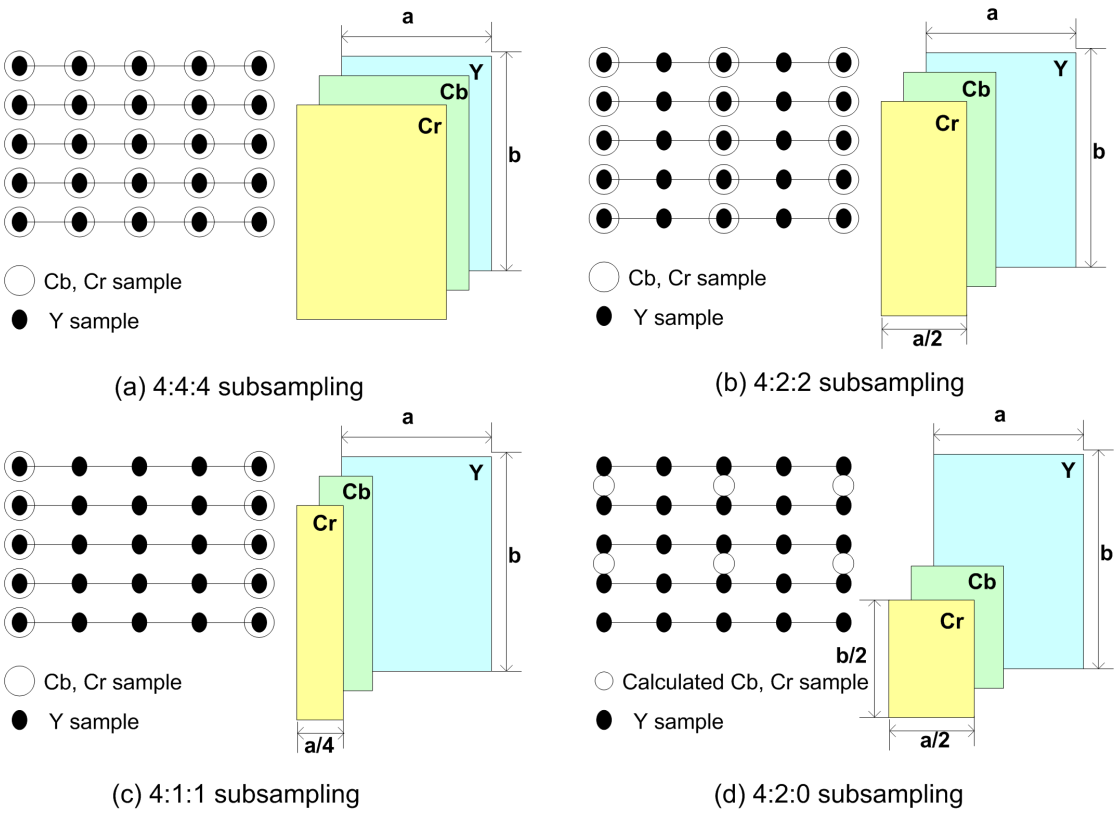


Figure 2.4: Colour subsampling (YCbCr format) in the case of progressive image

However, block based coding approach suffers from annoying blocking artifacts especially when used at low bit rates [47]. Two adjacent blocks may lose the original smoothness and continuity at the boundary. In spite of this problem, all video/image coding standards except for JPEG 2000 have used the block based hybrid coding because the block based video coding has shown excellent features; easy to implement and supply backward compatibility. Blocking artifacts can be overcome to some extend by applying an deblocking filter [95, 130]. We only overview block based video coding in this thesis.

2.3.1 Intra / Inter Prediction

Video coding has its own characteristics that make it quite different from still image compression. The major difference lies in the exploitation of inter-frame correlation that exists between successive frames in video sequences. In addition to it, the intra-frame correlation exists within each frame. The inter-frame

correlation is also referred to as temporal redundancy, while the intra-frame correlation is referred to as spatial redundancy. As far as video coding is concerned, two classes of techniques can be useful. The first class, which is also the most straightforward way to handle video coding, is to code each frame separately. Individual frames are coded independently. This is called "Intra Frame Coding" (I-Picture), where the target is the reduction of spatial redundancy. In the second class of techniques, several successive frames are grouped and coded together, referred to as "Inter Frame Coding" (B or P picture), whose ultimate goal is the reduction of temporal redundancy.

I, P and B Pictures (or Frame)

In I-Pictures, coding units are predicted using intra prediction without using previously coded pictures for prediction. These are used for the first picture of a sequence and random access pictures for reversing and forwarding. P-Pictures are inter predicted pictures with the reference as the nearest previously coded picture, which cannot be used as random acess due to the dependency on previously coded pictures. B-pictures are bi-directionally predicted pictures with two reference pictures (I or P-Pictures can be reference pictures), one from past and one from future in display order. They have high compression efficiency, however they are not used for reference or random acess pictures.

Intra Prediction

Intra prediction is a key function in intra frame coding. It has been commonly used to improve the coding efficiency in video coding. It utilizes the spatial correlation in an image to predict the block being encoded from its surrounding pixels. Spatial domain intra prediction was first introduced in H.264/AVC [123][1]. In the H.264/AVC intra coding, two intra MB modes for luminance are supported. One is Intra 4×4 prediction mode, and the other is Intra 16×16 prediction mode depicted as in Figure 2.5. Intra 8×8 is a new intra prediction type defined in H.264/AVC FREXt (see Chapter 4). For Intra 4×4, the MB is divided into 16 non-overlapping 4×4 luminance blocks and each 4×4 block can select one of

[1] There had been similar trials in the previous standards, MPEG-2 used a DC coefficient and, MPEG-4 Part 2 used several AC coefficients of neighboring blocks. However, these approaches represent not intra prediction but rather DPCM coding

nine prediction modes. For Intra 16×16, each MB can select one of the four modes. Chrominance intra prediction is independent of that of luminance. Two chrominance components are simultaneously predicted by one mode only. The possible chrominance prediction modes are very similar to those of Intra 16×16 except for different block size (8×8) and the index of DC mode. The Intra 4×4 mode can predict a block more accurately but requires more bits to represent the mode information than Intra 16×16. So, Intra 4×4 tends to be used for highly textured regions while Intra 16×16 tends to be chosen for plain regions. Mode decision is not specified in the H.264/AVC standard, but is arguably the most important step at the encoder side because it has an influence on coding efficiency. On the contrary, it consumes significant processing time and memory accesses.

From a complexity point of view, intra prediction finds the minimum cost by iterating intra decisions for each possible mode. Therefore, the number of mode combinations for luminance and chrominance components in a MB is given as

$$N_{mode} = C8 \times (L4 \times 16 + L16) \quad = 4 \times (9 \times 16 + 4) = 592. \tag{2.4}$$

Where $C8$, $L4$ and $L16$ represent the number of modes for chrominance prediction, 4×4 and 16×16 luminance prediction respectively. 592 different RDO calculations have to be performed before a best RDO mode is determined per MB.

Inter Prediction

Inter prediction is the main function of inter frame coding, which is a procedure to find temporal redundancy in successive frames. Early approaches to exploiting temporal redundancy may be traced back to the 1960s [94]. They presented a frame replenishment technique, where each pixel in a frame is classified into changing or unchanging areas between the current and the previous frame exceeds a threshold. For those unchanged pixels, nothing is coded. The major drawback of this technique is that it is difficult to handle frames containing more rapid changes (motions). ME and MC have been proved to be able to provide better performance than the replenishment technique in rapid change situations [90]. ME and MC coding have been used as a main tool to find temporal redundancy. In this technique, a motion model is assumed such

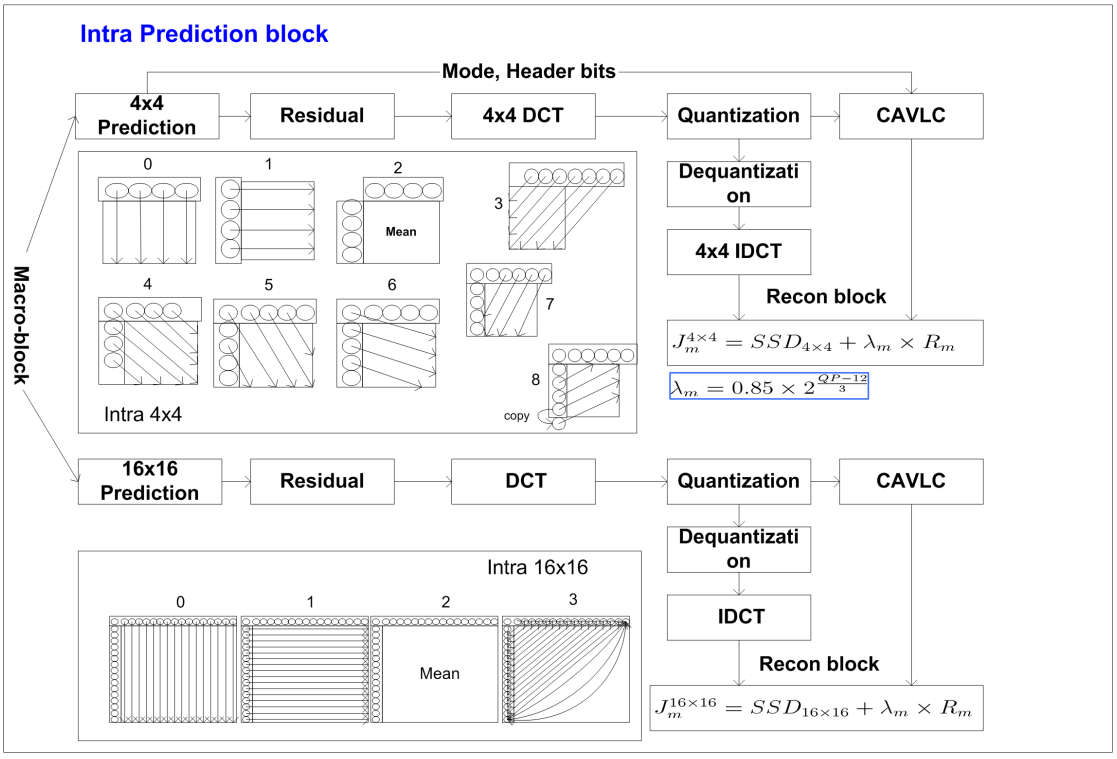


Figure 2.5: Illustration of the intra prediction process in H.264/AVC; Intra 4×4 consists of 9 modes, Intra 16×16 has 4 modes. The mode is chosen by finding minimum cost among $J_M^{4\times4}$ and $J_M^{16\times16}$

that the changes between successive frames are considered due to the translation of moving objects in the image plane. It consists of finding a displacement between the current and the previous frame, referred to as ME, and obtaining compensated frame referred to as MC. Figure 2.7 illustrates the inter prediction

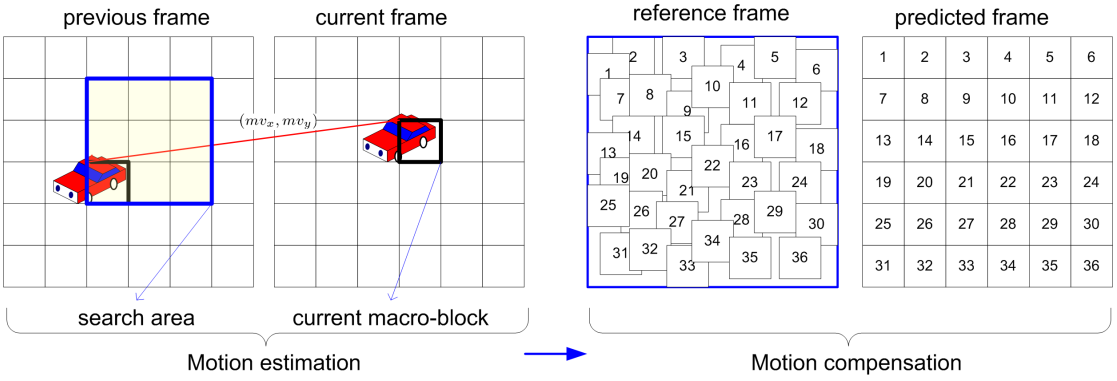


Figure 2.6: Illustration of inter prediction; motion vector (mv_x, mv_y) are obtained by taking minimum distortion between the current macro-block and search area. After motion compensation, a predicted frame is obtained.

procedure, which consists of both ME and MC. ME requires more computational complexity than MC. MC is a simple process to obtain a prediction frame if motion vectors are available via ME. A commonly used ME method is the Block Matching Algorithm (BMA), in which an image is partitioned into a set of nonoverlapped, equally spaced, fixed size and small rectangular blocks. The translation motion within each block is assumed to be uniform. This simple model is a quite good approximation for other types of motion in a small block size, including rotation and zooming. Motion vectors for blocks are estimated by finding their best matched counterparts in the previous frame. The simplest approach to find motion vectors is the Full Search (FS), where the correlation window is moved to each candidate position within the search area. It gives good accuracy in ME while a large amount of computational complexity is involved. In order to reduce computational complexity, fast searching algorithms have been developed. Fast algorithms are explained in Chapter 6. Although the BMA has been widely used in video coding because of its simplicity, straightforward method, and efficiency, it has drawbacks due to simple motion model as follows;

- The BMA mainly utilizes 2-D motion vector field, which is an unreliable motion vector field compared to true motion.
- The BMA needs to encode and transmit motion vectors as an overhead, so it is difficult to use smaller block size for accuracy.
- The BMA causes blocking artifacts at the boundary of blocks, which is especially severe at low bit rates.

Much research has been carried out to overcome the limitations of the BMA such as Overlapped Block Matching (OBM) [64], and Multiresolution Block Matching (MBM) [24]. Some improvements have been achieved. However, the BMA is still the most popular and efficient ME. Therefore, it has been adopted by almost all international coding standards.

2.3.2 Transform and Quantisation

Images, in their raw form, are not careless collections of arbitrary intensity transitions but embody some form of structure. As a result, there is correlation

between neighboring pixels. If one can find a reversible transformation that de-correlates data, an image can be coded more efficiently by quantisation of the data. Transform and quantisation is a necessary component in lossy coding and has a direct impact on the bit rate and the distortion of reconstructed images or videos.

Transform

Figure 2.7 depicts how an appropriate transform acts on correlated image data. The frame is divided into non-overlapping two adjacent pixel pairs as shown in Figure 2.7(a). In this example, we use the 1^{st} frame of the Mother and Daughter sequence with CIF (352×288) resolution. The scatter plot of these pairs is shown in Figure 2.7(b), where the strong diagonal relationship about the line clearly shows the strong correlation between neighbouring pixels. The distribution shown in Figure 2.7(b) is rotated by θ about the centre, then new axes (ϕ_1, ϕ_2) are depicted as follows;

$$\begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} \tag{2.5}$$

The two components are de-correlated, which means knowing the value of the first component (ϕ_1) does not help in estimating the value of the second (ϕ_2) as shown in Figure 2.7(c). ϕ_1 is still quite similar to the original distribution. On the contrary, ϕ_2 is quite different, it is much narrower with a peak at 0. Therefore, fewer bits are required to encode original values. This means that the number of bits required for encoding an image can be reduced by simple rotation of the axis.

The Karhunen Lòeve Transform (KLT) [43] has been shown to be the optimal transform in the sense of energy compaction, i.e. it places as much energy as possible in as few coefficients. It is a linear transform where the basis functions are taken from the statistics of the signal. Thus, it can be adaptive. The discrete version of KLT is also referred to as the Hotelling Transform (HT) or Principal Components Analysis (PCA). However, the KLT depends on the characteristics of the input signal. Whenever the characteristics of the signal change, eigen-vectors have to be recalculated (Transform matrix should be recalculated for different signal). A transform matrix may act on certain vectors by changing

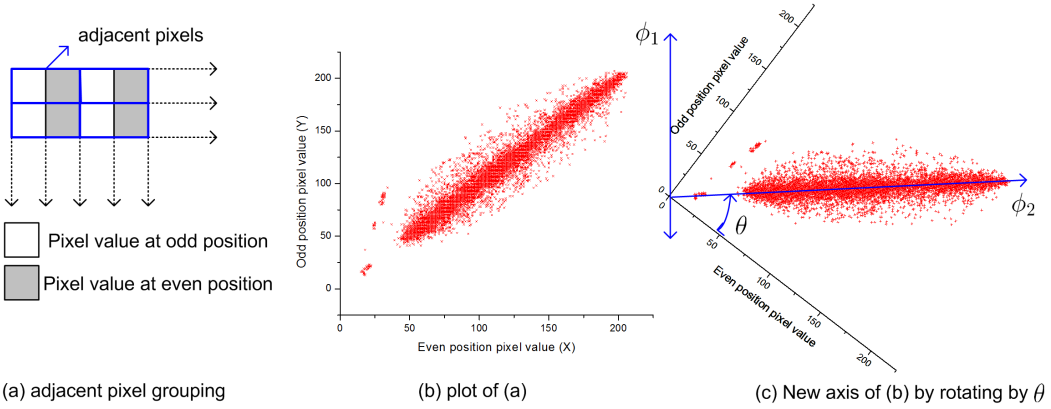


Figure 2.7: The principal of transform; test data are captured from 1^{st} frame of Mother and Daughter@352×288.

only their magnitude, and leaving their direction unchanged. These vectors are the eigenvectors of the transform matrix. Therefore, KLT is generally not feasible in real applications due to its computational complexity. Other Discrete Transforms (DTs) have been widely used instead of the KLT such as the Discrete Cosine Transform (DCT), the WHT, and so on due to their fast algorithms. The above mentioned DTs are called linear transforms, and provide the following benefits;

- Transform coefficients are less correlated than the original data.
- Some transform coefficients are more significant than others such that transform coefficients can be treated differently. Some coefficients could be discarded, coarsely quantised, or finely quantised.

Let a 2-D transform kernel or matrix be $\Phi(x, y, u, v)$ and its two 1-D transform kernel be Φ_1 and Φ_2, where (x, y) and (u, v) represents a 2-D data set in the pixel and transform domain respectively. Characteristics of linear transform are denoted as followings:

- **Separability**: A 2-D separable transform can be decomposed into two 1-D transforms as follow;

$$\Phi(x, y, u, v) = \Phi_1(x, u) \times \Phi_2(y, v). \tag{2.6}$$

- **Symmetry**: The transform is symmetric if the kernel is separable and the following condition is satisfied.

$$\Phi_1(x,u) = \Phi_2(x,u) \tag{2.7}$$

- **Unitary**: The transform is unitary satisfying the following condition, where Φ^{T*} is the conjugate transpose of Φ.

$$\Phi \times \Phi^{T*} = \Phi^{T*} \times \Phi = I \tag{2.8}$$

- **Orthogonality**: An orthogonal transform is a special case of a unitary transform, where only real values are involved.

Quantisation

Of course, simply transforming pixels does not actually yield compression. The energy in both the pixel and the transform domains are equal. However the majority of energy is concentrated on a few coefficients. Therefore, coefficients with little energy can be removed by quantisation. Moreover, by exploiting the human eye's characteristics, which are less sensitive to picture distortions at higher frequencies, one can apply even coarser quantisation at higher frequencies to give greater compression. Coarser quantisation step sizes force more coefficients to zero. As a result, more compression is gained. The principal of quantisation is the same as that of discretization as explained in Section 2.2.3. Note that the quantisation process of H.264/AVC is explained in Section 7.3.2 in detail.

2.3.3 Entropy Coding

The final step of an encoding system is entropy coding. Entropy encoding is a lossless data compression that is independent of the specific characteristics of the medium. One of the main types of entropy coding creates and assigns a unique prefix code to each unique symbol that occurs in the input. These entropy encoders compress data by replacing each fixed-length input symbol with the corresponding variable-length prefix code word. The length of each code word is approximately proportional to the probability. Therefore, the transform

coefficients and the coordinates of the motion vectors are entropy coded where short code words are assigned to the highly probable values and long code words to the less probable ones. Entropy coding commonly consists of Run-Length Coding (RLC) and Variable Length Coding (VLC). Here we provide an overview of an Content-Adaptive Variable Length Coding (CAVLC) which is one entropy coder used in H.264/AVC. CAVLC is used as a mandatory tool for the RDO procedure and thus related to the work on measuring bit-rates in this thesis.

CAVLC uses RLC to present strings of zeros compactly. CAVLC is developed based on several observations; (1) The highest nonzero coefficients after zigzag scan are often ± 1 called "Trailing Ones". (2) The number of nonzero coefficients in neighbouring blocks are correlated. (3) The magnitude of nonzero coefficients, called "Level", tend to be larger near the DC coefficient and smaller towards the high frequencies.

An example of CAVLC is shown in Figure 2.8 and Table 2.1. Figure 2.8 shows an example block and its information to be used in CAVLC encoding, whose meanings are denoted as following;

- **run_before**: The number of zeros before a specific non-zero DCT coefficient.
- **Level**: Magnitude of DCT coefficient.
- **TrailingOnes**: Total number of ± 1.
- **Totalcoeffs**: Total number of non-zero DCT coefficients.
- **total_zeros**: Total zeros before last non-zero DCT coefficient.

Run and level are individually encoded using VLC tables in inverse order of zigzag scan. Other information except run and level is adaptively encoded using neighbouring MBs by selecting an appropriate VLC table. This is one of the reasons why CAVLC shows good coding efficiency over other VLCs. The VLC table can be selected as an average of the number of DCT coefficients of a left and a upper block although there is no regulation for which VLC table is used.

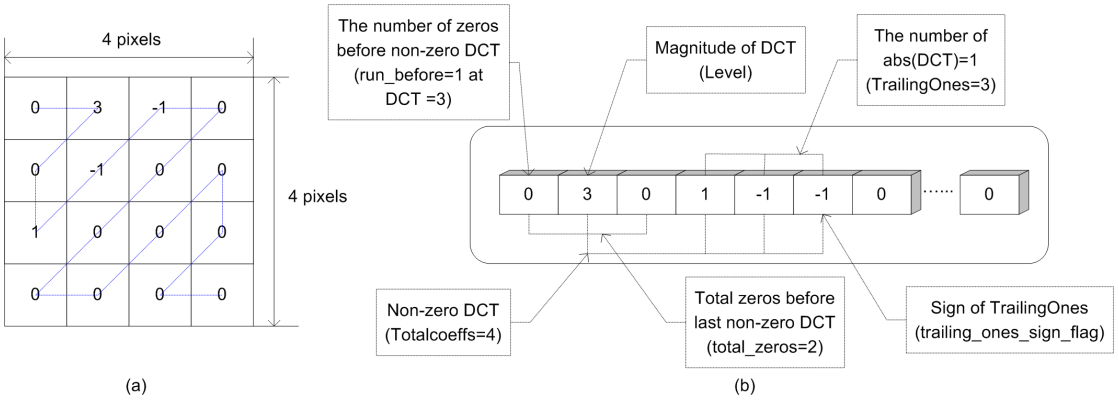


Figure 2.8: Notation of CAVLC example; (a) DCT coefficients of a 4×4 block. (b) Information of CAVLC encoding

Table 2.1: Example of encoding CAVLC with Figure 2.8

Element	Value	Code
coeff_token	TotalCoeffs=5	0000100
TrailingOnes =3 (Table 1)		
TrailingOne sign (1)	+	0
TrailingOne sign (-1)	-	1
TrailingOne sign (-1)	-	1
Level (1)	+1 (suffixLength=0)	1 (prefix)
Level (3)	+3 (suffixLength=1)	001(prefix)+0(suffix)
total_zeros	3	111
run_before(1)	ZerosLeft=3, run_before=1	10
run_before(-1)	ZerosLeft=2, run_before=0	1
run_before(-1)	ZerosLeft=2, run_before=0	1
run_before(1)	ZerosLeft=2, run_before=1	01
run_before(3)	ZerosLeft=1, run_before=1	No coded.
Encoded bitstream :	000010001110010111101101	

2.4 Conclusion

Image/video coding is a process in which the amount of data are reduced to meet a bit rate or complexity requirement for a given condition. In this chapter, key functionalities of the digital video coding are briefly overviewed. Prediction plays a key role in reducing spatial or temporal redundancy. Transformation is a series process to de-correlate the original correlated data, where quantisation acts on removing less important information based on the fact that the HVS is less sensitive to high frequency components. Finally, entropy coding based on information theory makes use of the probability of occurrence to reduce redundancy.

—You will never find time for anything. If you want time you must make it.

Charles Buxton

3 Experimental Method Used for Complexity Adaptation in a Video Coder

3.1 Introduction

Low complexity in an encoder is especially useful for applications operating on power limited platforms such as a wireless camera network and mobile devices. The contribution of the complexity adaptation algorithm is to enable control of the computational cost while ensuring a low complexity encoder. The proposed complexity adaptation framework measures the level of complexity by controlling the number of skipped macro-blocks. Computational savings are achieved by early prediction of skipped macro-blocks prior to time consuming functions of an encoder.

This chapter presents the experimental method used to design low complexity sub function blocks and the complexity adaptation framework. The design and validation of the complexity adaptation framework is presented in Chapter 8. This is the ultimate goal of this thesis.

3.2 Experimental method

3.2.1 Test Sequences

The test video sequences used are mostly chosen from widely used test material in video coding research. There include different foreground and backgrounds,

motion, detail, blurring, sharpening, and camera movements from resolutions of 176×144 to 1280×720. Table 3.1 shows the characteristics of the test sequences. Sample frames and their features are briefly overviewed in the following.

Table 3.1: Test sequences

Sequence title	Foreman	Mother&Daughter	Rush hour	Pedestrian	Blue sky
Resolution	176×144, 352×288		720×576, 1280×720		
Number of frames	300		100		
Colour space	4:2:0 YUV				
Frames per second	30		25		
Source	Uncompressed, progressive				

Foreman

(a) (b) (c)

Figure 3.1: Foreman sequence; (a) 1^{st}, (b) 100^{th}, (c) 200^{th} frame

This is one of the most well-known sequences. It includes a face with a very rich variety of expression. The motion that is present is disordered and does not have any forward characteristics. The complexity of the motion creates problems for the motion compensation process. Moreover, the camera is shaking, which makes the image unsteady. The camera suddenly turns to the building site at the end of the sequence. Therefore, this sequence can be used to test the behavior of the codec for a static scene followed by one with motion.

Mother and Daughter

This is a low complexity head and shoulders sequence with moderate amount of detail. The camera is static with moderate movement. Moreover, large moving areas correspond to shaking heads. This sequence is representative of static camera applications such as surveillance or telephony.

Figure 3.2: Mother& Daughter sequence; (a) 1^{st}, (b) 100^{th}, (c) 200^{th} frame

Rush Hour

This correspond to a traffic scene in rush hour. It has plenty of large motion areas, and an image with refraction due to haze. Vehicles pass by close to the camera in both forward and backward direction. This sequence is representative for compression of complex moving objects.

Figure 3.3: Rush hour sequence; (a) 1^{st}, (b) 50^{th} frame

Pedestrian

This is a shot of a pedestrian area. This sequence has a low camera position, people pass very close to the camera, with high depth of field and a static camera. It is a suitably challenging rear sequence for compression, due to the static camera and areas of different focusing, blurring, and sharpening.

(a) (b)

Figure 3.4: Pedestrian sequence; (a) 1^{st}, (b) 50^{th} frame

Blue Sky

This is a shot of blue sky aimed directed at a tall tree. This sequence has a camera rotation. The tree has a moderate amount of detail and camera rotation generates a large global motion. The most common fast ME assumes that dissimilarity monotonically increases as the search point moves away from the point corresponding to the minimum dissimilarity. Fast ME is easily trapped in a local minimum, which frequently occurs in a camera rotate.

(a) (b)

Figure 3.5: Blue sky sequence; (a) 1^{st}, (b) 50^{th} frame

3.2.2 Objective Test Metrics Description

Peak Signal-to-Noise Ratio (PSNR)

PSNR is often used in practice as a quality measure and its definition is given by

$$PSNR = 10 \log_{10} \left(\frac{255^2 \times N}{\sum_{i=1}^{N} (X_i - Y_i)^2} \right) \tag{3.1}$$

where X_i and Y_i represent the pixel value for the i^{th} position in frames X and Y. N is the as total frame size. This metric has the same form as the Mean Squared Error (MSE). However, it is more convenient to use due to its logarithmic scale. It is sometimes inappropriate in terms of how it relates to the HVS. For example, not all sequences that generate a high PSNR give good subjective quality.

Bjontegarrd's Delta Peak Signal to Noise (BDPSNR) and Bjontegarrd's Delta Bit-rate (BDBR)

The difference of two R-D curves can be displayed using BDPSNR and BDBR [9] as shown in Figure 3.6. In this case, the algorithm under evaluation is compared to an anchor reference, then numerical averages between the R-D curves are obtained via BDPSNR and BDBR. This is a more compact and accurate way to represent R-D performance. Therefore, no distinction between total range and local range is needed. From [9], the relationship between ΔSNR and ΔBit-rate is well represented by $0.5dB = 10\%$ or $0.05dB = 1\%$. Therefore, the measurement of R-D performance can be obtained to calculate either change in bit rates or change in PSNR. Interpolated logarithmic bit-rate can be calculated with a third order polynomial form given by

$$PSNR = a + b \times bit + c \times bit^2 + d \times bit^3 \tag{3.2}$$

where a, b, c, and d are fitting constants. After obtaining interpolated curves, the average of PSNR is obtained by comparing by integrating the area of each curve.

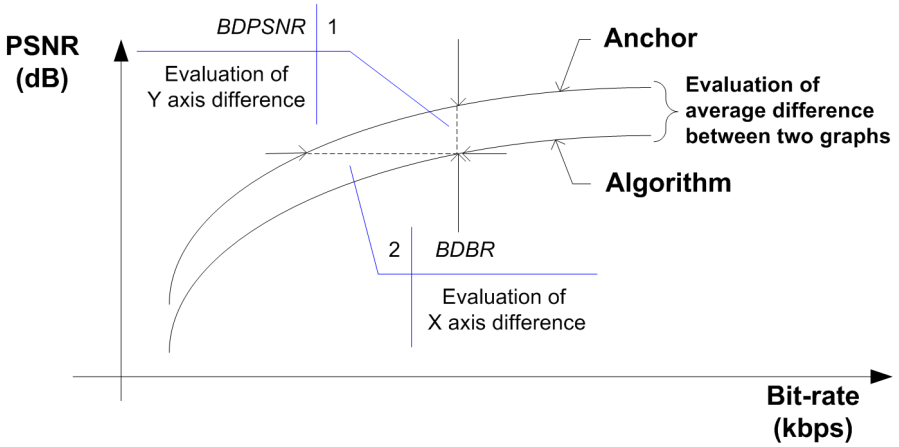


Figure 3.6: The meaning of BDPSNR and BDBR

SSIM [122, 134]

This metric was presented motivated by the drawback of PSNR. The main idea that underlies the Structural SIMilarity (SSIM) index is comparison of the distortion of three image components; (1) Luminance. (2) Contrast. (3) Structure. The SSIM can be obtained via the following expression.

$$SSIM(X,Y) = \frac{(2\mu_X\mu_Y + C_1)(2\sigma_{XY} + C_2)}{(\mu_X + \mu_Y + C_1)(\sigma_X + \sigma_Y + C_2)} \tag{3.3}$$

where

$$\mu_X = \sum_{i=1}^{N} \omega_i X_i, \quad \mu_Y = \sum_{i=1}^{N} \omega_i Y_i$$

$$\sigma_X = \left(\sum_{i=1}^{N} \omega_i (X_i - \mu_X) \right)^{\frac{1}{2}}, \quad \sigma_Y = \left(\sum_{i=1}^{N} \omega_i (Y_i - \mu_Y) \right)^{\frac{1}{2}}$$

$$\sigma_{XY} = \sum_{i=1}^{N} \omega_i (X_i - \mu_X)(Y_i - \mu_Y).$$

The constants C_1 and C_2 are defined as;

$$\begin{aligned} C_1 &= (K_1 L)^2 \\ C_2 &= (K_2 L)^2 \end{aligned} \tag{3.4}$$

where L is 255 if 8-bit gray scale images are used. K_1, K_2 are constants reasonably smaller than 1, which are selected as $K = 0.01$ in most cases. The SSIM value corresponds to two sequences and its range is $[-1, 1]$. One of the advantages of the SSIM metric is that it better represents the HVS than PSNR. However it takes more time to calculate.

Complexity measure metrics

In Chapter 1, we have outlined a parametric video encoding whose complexity is fully affected by the number of SAD and skip macroblocks. To translate the complexity into energy, we need to consider the energy in hardware design. To dynamically control the energy consumption of the microprocessor on the portable device, a CMOS circuits design technology, Dynamic Voltage Scaling (DVS) has been developed [67]. In [32], lowering the supply voltage will reduce the energy consumption of the system which is given by

$$P \propto f_{clk}^3 \tag{3.5}$$

where f_{clk} is the clock frequency of the circuit. It can be seen that the CPU can reduce its energy consumption substantially by running more slowly.

Therefore, a complexity metric can be modeled by measuring the running frequency in the middle of the encoding processing. It can be simply obtained by reading special register of the CPU. In this thesis, we use both running frequency and total encoding time as a complexity metric. The former metric can be used in the middle of encoding, the latter one is used after encoding process.

3.2.3 Simulation Models

As part of our experimental procedure we use two software simulators; (1) a modified JM and (2) a complexity adaptation framework as shown in Figure 3.7. In the modified JM, low complexity sub functions are integrated into JM[1] (see Chapter 4). In a modified JM, the VBS partitioning algorithm and fast ME based on SWHT called FWBS are used as sub functions. Details are explained in Chapter 5 and Chapter 6. In the complexity adaptation algorithm, skip MB detection algorithm is proposed in order not to have to perform ME, ICT, quantisation, and entropy coding. Finally, the complexity control algorithm for adjusting the threshold value of skip MB is presented in Chapter 8. The R-D and C-D performance of Figure 3.7(b) is compared with the JM reference software (see Figure 3.7(a)). The performance of the complexity adaptation

[1] JM is the abbreviation of Joint Model, which is JVT reference software for H.264/AVC. Source code is available at http://iphome.hhi.de/suehring/tml/

framework (see Figure 3.7(c)) including low complexity sub functions such as VBS , FWBS, and skip MB detection algorithm is compared to that of the modified JM (Figure 3.7(b)).

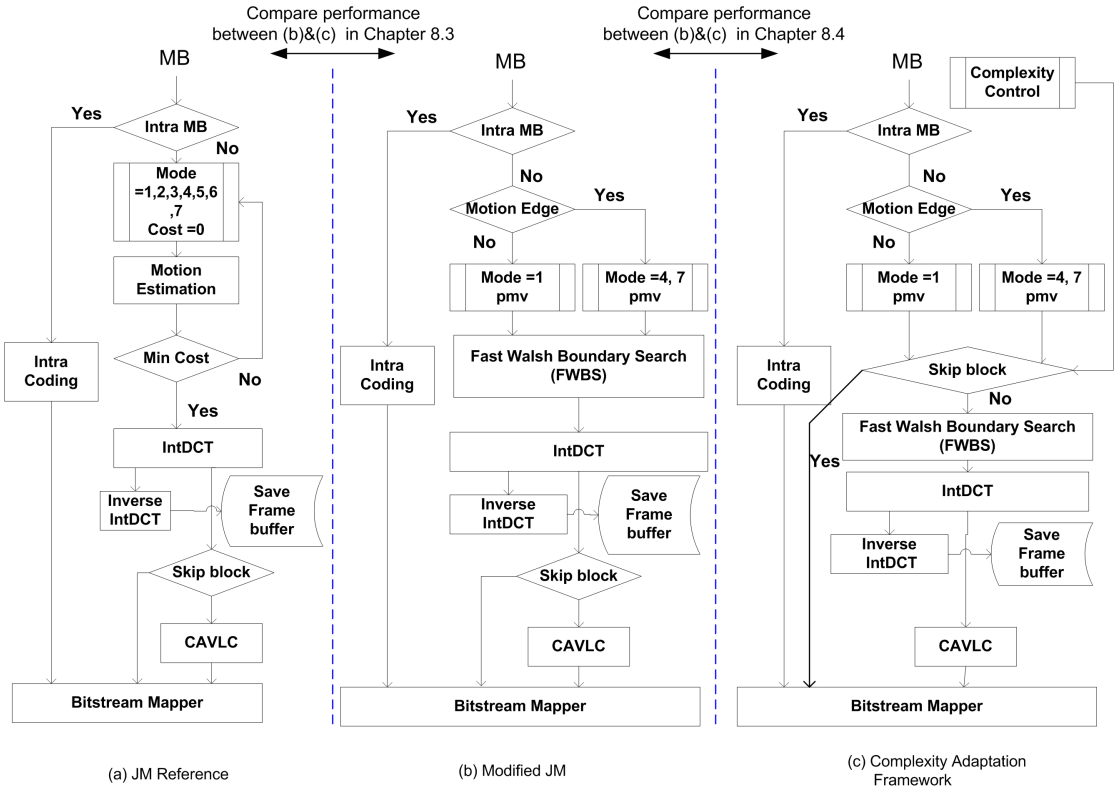


Figure 3.7: Flowchart of three different schemes to compare performance; (a) Functional flowchart of JM 11.0 reference software, (b) JM with VBS and FWBS, (c) Complexity control algorithm. The different R-D and C-D performances between (a)&(b) is discussed in Section 8.3, (c) is compared with the modified JM (b).

3.3 Discussion

This chapter describes the test sequences, objective quality metrics, and the experimental method as used in this thesis. Test sequences are carefully selected from QCIF to HD considering the presence of motion, detail, foreground, and background. The first experimental procedure consists of testing the modified JM with proposed new sub functions such as VBS and fast ME. The next experiment includes skip MB detection and a complexity adaptation algorithm in the modified JM. Figure 3.7 shows the flowchart of the three different frameworks,

where the modified JM acts as an intermediate step to reach the complexity adaptation framework. An overview of complexity issues associated with H.264/AVC is presented in the next chapter prior to presenting the key contributions of this thesis.

—Success is not a place at which one arrives but rather the spirit with which one undertakes and continues the journey.

Alex Noble

4 An Overview of H.264/AVC : Complexity Perspective

4.1 Introduction

Joint effort by ITU-T's Video Coding Experts Group and ISO/IEC's Moving Picture Experts Group resulted in standardization for H.264/AVC in 2003 [123]. The specification of the encoder is not defined because standardization has focused on only a decoder. Therefore, there has been various H.264/AVC encoders developed by individuals or organizations[1]. The encoder developed by the Joint Video Team, known as the Joint Model (JM), has been used as a reference by encoder developers in enhancing existing algorithms. However, its use has been limited due to its encoding speed. Another H.264/AVC open source encoder is the x264 [3]. It has been used in many applications like ffmpeg [1], MEcoder, and ffdshow. In a recent study, x264 showed better quality than several commercial H.264/AVC encoders [53]. Moreover, x264 shows fast encoding time by a factor of ten over JM [73]. The high performance in terms of encoding time is attributed to its algorithms (rate control, motion estimation, and mode decision), called "algorithmic optimization" process, and optimized code for many of the primitive operation using assemble code on a specific platform,"platform optimization". In the latter case, it is very platform-specific, which normally produces the most efficient code since optimization can take advantage of the full repertoire of machine instruction. However, platform optimization causes several problems; (1) Development time is much longer than

[1] In [53], various performance comparisons are performed on various H.264/AVC encoders

in a high level approach. (2) It is easy to make errors, which affects the reliability and security. (3) Porting to a different platform is difficult. In the former case, it is the optimization methods according to their algorithmic structure and underlying principles from the viewpoint of theory, which gives a robust and platform-independent solution. Therefore, in this thesis we consider a non-optimized H.264/AVC encoder to show how much improvement is achieved by applying the proposed algorithms in terms of complexity. In the following sections, an overview of H.264/AVC encoding process is suggested based on an observation of which H.264/AVC encoding tools contribute most to the complexity and bit-rates.

4.2 H.264/AVC standards

4.2.1 Standards History

In 1998, a call for proposals was issued by ITU-T Video Coding Expert Group (VCEG) for a new video coding standard with the objective of doubling the compression efficiency compared to previously existing video coding standards. The new proposal was known to as H.26L. However, as a result of similar interest by ISO/IEC Moving Picture Experts group (MPEG), the Joint Video Team (JVT), consisting of ITU-T VCEG and MPEG was formed in 2001 to develop the new standard. The standard was finalised and the draft was approved in 2003.

The H.264/AVC standard was originally developed for “entertainment quality” video where sampling format is limited to 4:2:0 with 8 bit sample accuracy. An amendment was added to the standard in July 2004 called the Fidelity Range Extensions (FRExt) [104], where “High Profiles" were provided in order to address professional applications and to enhance the compression performance. The high profiles can support up to 4:4:4 sampling format and 12 bit sample accuracy. Moreover, an advanced 4:4:4 Profile has been proposed to code 4:4:4 format video which includes coding of chroma components in 4:4:4 with luma coding tools and is reported to outperforme the High 4:4:4 profile.

The H.264/AVC standard was designed for high compression efficiency, error resilience and flexibility so that it could support a wide variety of applications and different transport environments such as wired and wireless networks.

4.2.2 Features of H.264/AVC

Layer Structure

H.264/AVC was designed to be flexible and customizable to handle a variety of applications and transport methods. To achieve the flexibility, the standard contains two layers.

1. The Video Coding Layer (VCL) represents the core video encoding process (which carries out actual video compression) and the VCL data consists of coded bits.
2. The Network Abstraction Layer (NAL) handles the transportation of VCL data and other header information by encapsulating them in NAL units.

The separation of video coding and transportation into two layers ensures that the video coding layer provides an efficient representation of video content, while the network abstraction layer transports the coded data and other header information in a flexible manner by adapting to a variety of delivery frameworks.

Profiles and Levels

The standard includes the following set of capabilities referred to as profiles, which target specific classes of applications;

- **Baseline Profile (BP):** This is for lower cost applications with limited resources. This profile is widely used in mobile applications.
- **Main Profile (MP):** This profile was intended for the mainstream consumer for broadcast and storage applications.

- **Extended Profile (XP):** Profile was intended for the streaming video. It supports relatively high compression capability and some extra tricks for robustness to data losses.

In addition, new profiles were introduced in FREext:

- **High Profile (HiP):** This was introduced for High-Definition (HD) Television applications both broadcast and disc storage applications.
- **High 10 Profile (Hi10P):** This profile built on top of the HiP by adding support for up to 10bits per sample.
- **High 4:2:2 Profile (Hi422P):** This targets professional applications that use interlaced video by adding 4:2:2 chroma subsampling format using 10bits per sample.
- **High 4:4:4 Predictive Profile (Hi444PP):** This profile built on top of the Hi422P. This supports 4:4:4 chroma subsampling.

Moreover, in order to support professional applications such as camera and editing system, the standards contains additional all intra profiles:

- **High 10 Intra Profile:** It is constrained to only intra use in Hi10P.
- **High 4:2:2 Intra Profile:** It is constrained to only intra use in Hi422P.
- **High 4:4:4 Intra Profile:** It is constrained to only intra use in Hi444PP.
- **CAVLC 4:4:4 Intra Profile:** It is constrained to only intra use in Hi444PP. However it does not support CABAC.

As a result of Scalable Video Coding extension [93], the standards contains additional scalable profiles defined as a combination of the H.264/AVC profile;

- **Scalable Baseline Profile:** This profile targets for video conferencing, mobile, and surveillance applications.
- **Scalable High Profile:** This profile targets for broadcast and streaming applications based on HiP.

- **Scalable High Intra Profile:** This profile is constrained to all intra use only in Scalable High Profile.

H.264/AVC has 11 levels or degree of capability to limit performance, bandwidth and memory requirements. Each level defines the bit rate and the encoding rate in macroblock per second for resolutions ranging from QCIF to HDTV and beyond. The higher the resolution, the higher the level required.

In this thesis, all tests are performed in the Baseline Profile which is target for limited resources, however levels are used from QCIF (Level 1.x) to HD (Level 4.x).

4.3 Contributive Factors to H.264/AVC Complexity

4.3.1 H.264/AVC Encoder Functionalities at Macro-block Level

Figure 4.1 shows the block diagram of an H.264/AVC encoder at macro-block level. A macro-block is divided into smaller partitions for VBS ME. For luminance, block sizes of 16×16, 16×8, 8×16, and 8×8 samples are supported, where mode numbers are allocated from 0 to 4. Note that mode 0 means a skip macro-block. In case of an 8×8 sub macro-block, the corresponding 8×8 is further divided into partitions with block sizes of 8×4, 4×8, and 4×4, where modes numbers are also allocated 5 to 8. In addition to those block sizes, two more modes are needed to make Mode Decision (MD), which are 16×16 and 4×4 intra blocks because each macro-block can be encoded in intra or inter mode. Therefore, MD controls a combination of 8 different block sizes and 2 intra block sizes. In intra mode, prediction is formed from samples from macro-blocks that have been previously encoded, decoded and reconstructed in the current frame. In inter mode, motion prediction (pmv) is obtained via neighbouring blocks' motion vector due to motion vectors close relationship with spatial correlation. Prediction is formed by ME and MC from one or more reference frame (RF) because H.264/AVC supports multi-frame ME and MC. The prediction is subtracted from the current macro-block to produce a residual (R). This is

transformed and quantised to give a set of quantised transform coefficients, together with side information to decode the macro-block such as macro-block motion prediction and motion vectors. The quantised macro-block coefficients are decoded in order to reconstruct a frame for encoding of further macro-blocks. That is, the coefficients undergo inverse quantisation (IQ) and inverse integer DCT. After MC, macro-block reconstructed (MBR) data is obtained. A filter is applied to reduce the effects of blocking distortion and saved to RF buffers.

In the MD process, the above procedure is performed on different modes. If only luminance is considered, the number of modes is given by

$$\begin{aligned} N_{mode} &= \overbrace{4}^{\text{Inter Mode (1-4)}} + \overbrace{2}^{\text{Intra}} + \overbrace{4 \times 4 \times 4 \times 4}^{\text{Inter sub block combinations}} \\ &= 262. \end{aligned} \tag{4.1}$$

Therefore, MD finds the argument i to satisfy the minimum value of Equation (4.2) of all combinations.

$$MB* = \arg\min_{i}(J_i(QP) = D_i(QP) + \lambda R_i(QP)) \tag{4.2}$$

It should be noted that only CAVLC is used to calculate $R_i(QP)$ in MD because Context-Adaptive Binary Arithmetic Coding (CABAC) requires huge computational complexity compared to CAVLC. Finally, the $MB*$, which is a optimum mode, is obtained.

4.3.2 Contributive Factors to Complexity

This research is aimed at a complexity adaptation framework which mandatorily requires low complexity functionality. In order to address computational complexity issues, we first need to know which functions generate computational complexity by understanding the complexity of an encoder.

From Figure 4.2, MD requires the most complexity in an encoder, and ME is the second most demanding. Moreover, ME is a sub function of MD because it uses a ME. The above two functions occupy almost 79% in "Foreman" sequence. When sub-pel ME is used, MC could be a part of motion estimation to obtain sub-pel motion vector. Therefore, almost all complexity related factors have a very close relation with mode decision and ME. Moreover, in a decoder, MC is

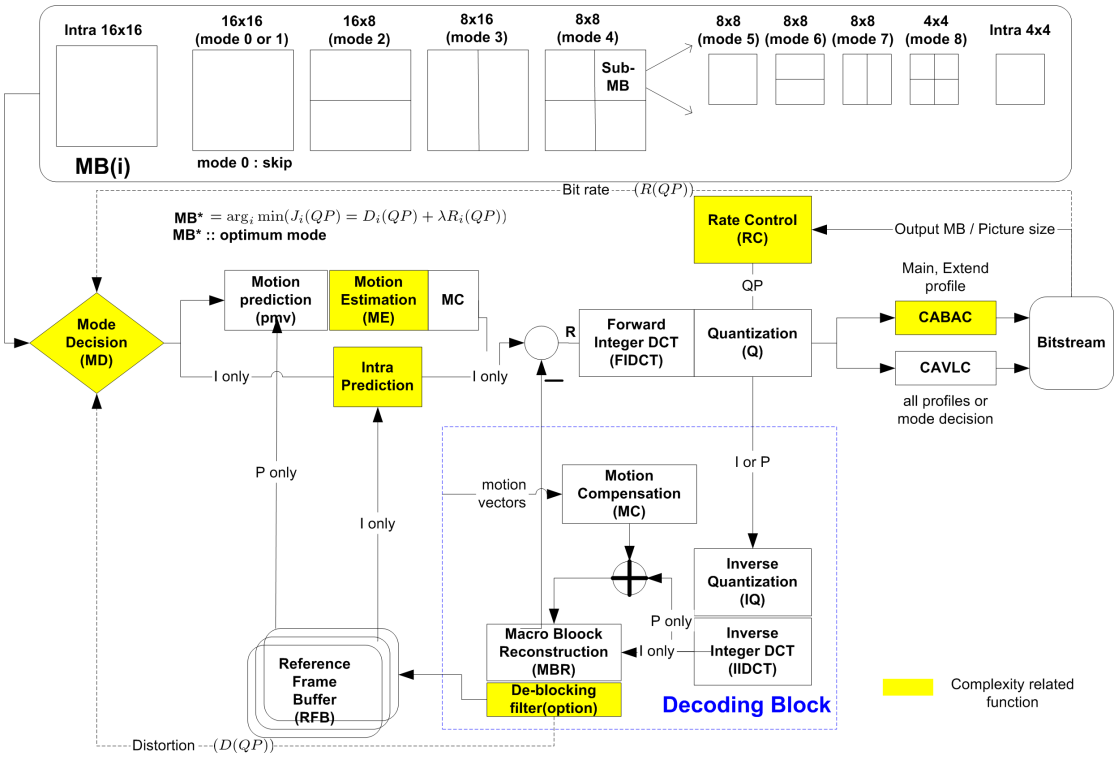
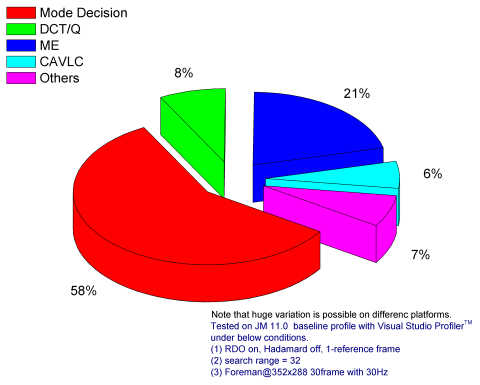
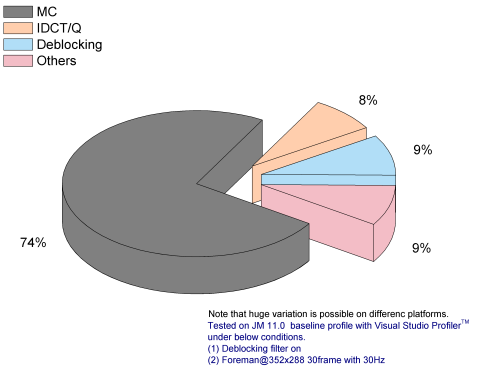


Figure 4.1: H.264/AVC functional block diagram; marked yellow functions require high computational complexity in the H.264/AVC encoder



(a) H.264/AVC encoder computational costs; mode decision and ME occupy 79%



(b) H.264/AVC decoder computational costs; MC is the most computationally demanding block

Figure 4.2: Computational costs of H.264/AVC tools

also the most complexity demanding function. We would like to observe their characteristics and affect on R-D and computational complexity. These factors can be classified as follows; (1) different VBS partitioned ME, (2) sub-pel motion vector resolution, (3) various search range, (4) the number of multiple reference frames, (5) existence of RDO algorithm, (6) presence of Hadamard Transform, and (7) CABAC[2]. Test conditions and their summarized procedures are denoted in Table 4.1.

Table 4.1: Test conditions and procedures of contributive functions on computational complexity of an encoder

Test Conditions		
Sequences	Foreman, Mother & Daughter @352x288 30Hz	
GOP	IPPP structure, I-frame at every 30frame	
Evaluation	Average PSNR, Bit-rate (BDPSNR, BDBR),and normalized complexity	
Test Procedure		
	JM 11.0 Anchor (Baseline profile)	Modified JM with Encoding Parameters
VBS	All modes	(1) Group 1 : 16×16 (mode 0,1) (2) Group 2 : $16 \times 16, 8 \times 16, 16 \times 8, 8 \times 8$
RDO	RDO on	RDO off
Hadamard	Hadamard on	Hadamard off
Sub-pel ME	$\frac{1}{4}$-pel accuracy	Integer-pel accuracy
Search Range (SR)	± 16	(1) ± 8 (2) ± 4
Multiple reference frame (FR)	5	(1) 3 (2) 1
CAVLC / CABAC	CABAC	CAVLC

VBS

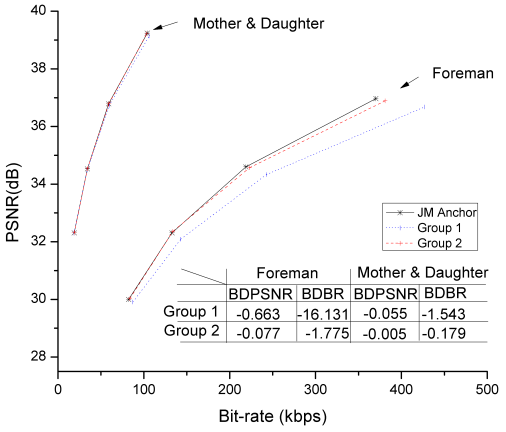
H.264/AVC supports VBS ME, where macro-blocks are partitioned into different block sizes. However, to achieve high compression, the encoder needs to evaluate all possible combinations of block size resulting in high computational complexity. In order to evaluate the effect of VBS on R-D and C-D performance, macro-block partition modes are grouped into two mode groups (see rightmost column of Table 4.1) and sequences are encoded using each mode group. Figure 4.3 shows the R-D and C-D performance of the macro-block partition mode groups. According to the results, compression efficiency is improved as the number of macro-block modes evaluated is increased. The BDPSNR, which is mentioned in Section 3.2.2, of only 16×16 decreases by 0.663dB for "Foreman" and by 0.055dB

[2] CABAC is not used in this thesis, but its effect on performance is observed because it is one of complexity required functions in the H.264/AVC encoder

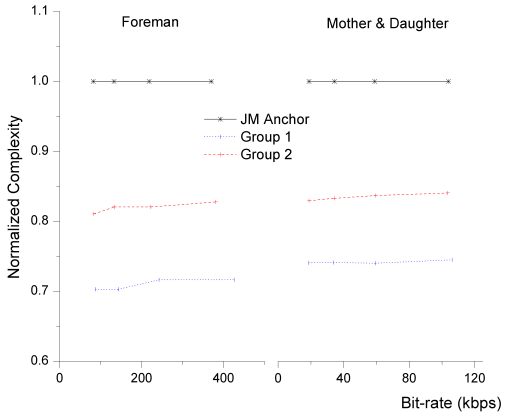
relative to using all modes for "Mother and Daughter". It can be evaluated as a equivalent index to the difference in R-D performance as follows [9];

$$\Delta bitrate = 20 \times BDPSNR. \tag{4.3}$$

Therefore, bit-rate increases by 13.26% and 1.1% for "Foreman" and "Mother and Daughter" respectively. The VBS significantly affects R-D performance in a sequence which has large motions such as "Foreman". However, it does not have much of an affect for a sequence that has relatively small motions or static scenes such as "Mother and Daughter". In the case of another test on VBS partition Group 2 (not using sub macro-block modes), increase in bit-rate is not significant for both sequences (less than 1.5%). The normalized complexity is reduced by 30% for 16×16 only mode (Group1) and 20% when not using sub modes (Group 2).



(a) R-D performance; R-D degradation is more severe in Foreman than Mother and Daughter



(b) C-D performance; Group 1 and Group 2 requires about 70%. 80% complexity compared to JM which uses all modes

Figure 4.3: Rate-distortion performance and normalized complexity ($\frac{Targetcomplexity}{JMcomplexity}$) of different macro-block partition mode groups

In conclusion, VBS should be considered in the case of heavy or large motion sequences in order to improve R-D performance.

Sub-pel motion vector resolution

Sub-pixel ME and MC plays an important role in compression efficiency within modern video codecs such as MPEG-2, MPEG-4 and H.264/AVC. Sub-pixel

motion compensation is implemented within these standards using interpolated pixel values at half-pel or quarter-pel accuracy. Such interpolation gives a good reduction in residual energy for each predicted macro-block and therefore improves compression. However, such interpolation is very computationally complex for the encoder. This is especially true for H.264/AVC where the cost of an exhaustive set of macro-block segmentations need to be estimated in order to obtain an optimal mode for prediction. Therefore, we would like to observe the effect of sub-pel ME and MC on both R-D and C-D performance in this section. JM performs ME by separating integer-pel and sub-pel accuracy in order to reduce complexity. In the integer-pel ME, predicted motion vectors (pmv) are obtained taking median value of neighbouring blocksŠ motion vectors and the centre of searching area is set as a pmv not a absolute position (0,0). Spiral searching is performed from the pmv position to the whole search range.

Sub-pel accuracy ME is performed after finding a location that generates minimum cost in the integer-pel position. The procedure of sub-pel ME consists of two steps. Firstly, let E have minimum cost resulting from integer-pel ME at [A–I] as depicted in Figure 4.4(a). Then, let the position that indicates minimum cost out of nine half-pel positions denoted as [1–9] around E be 6. As the same as finding half-pel motion vectors, the optimal motion vector can be obtained by comparing cost of 6 with those of nine quarter-pel positions represented [a–h]. Secondly, MC interpolation procedure is needed to generate pixel values of sub-pel positions as shown in Figure 4.4(b). In the luminance component, the sub-pel samples at half-pel positions (denoted as under bar numbers) are generated first and are interpolated from neighbouring integer-pel samples using a 6-tap Finite Impulse Response (FIR) filter. This means that each half-pel sample is a weighted sum of 6 neighbouring integer samples. For example, $\underline{3}$ is obtained via FIR filtering using horizontal 6 integer-pels [E–J], which is denoted as (1) in Figure 4.4(b), in the following.

$$\underline{3} = (E - 5 \times F + 20 \times G + 20 \times H - 5 \times I + J)/32^{3} \tag{4.4}$$

Moreover, $\underline{10}$ is interpolated using horizontal 6 integer-pels [A–S], which is denoted as (2) in Figure 4.4(b).

$$\underline{10} = (A - 5 \times C + 20 \times G + 20 \times M - 5 \times Q + S)/32 \tag{4.5}$$

[3] Shift operartion is used in JM considering rounding. $\frac{X}{2^n} = (X + 2^{n-1}) >> n$.

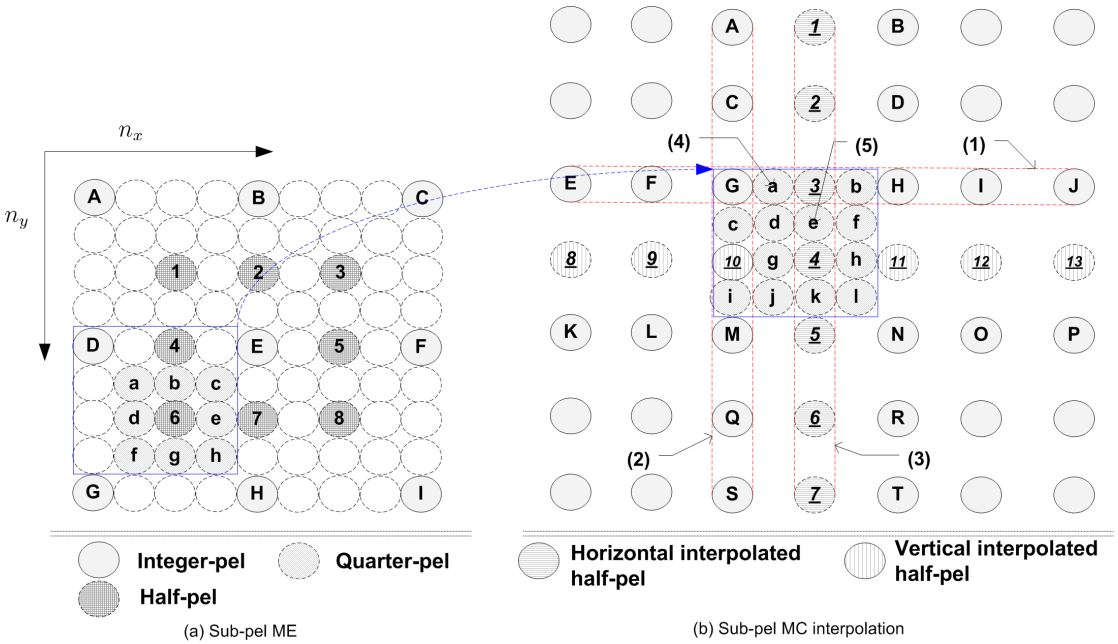


Figure 4.4: Illustration of sub-pel ME and MC interpolation; upper-case letters indicate the integer-pel position, under bar numbers or numbers indicate half-pel positions and remaining lower-case letters indicate quarter-pel positions

The half-pel position ($\underline{4}$) located in 4 integer-pels can be obtained via 6-tap FIR filtering in both horizontal or vertical directions shown in the following (see (3) in Figure 4.4(b)).

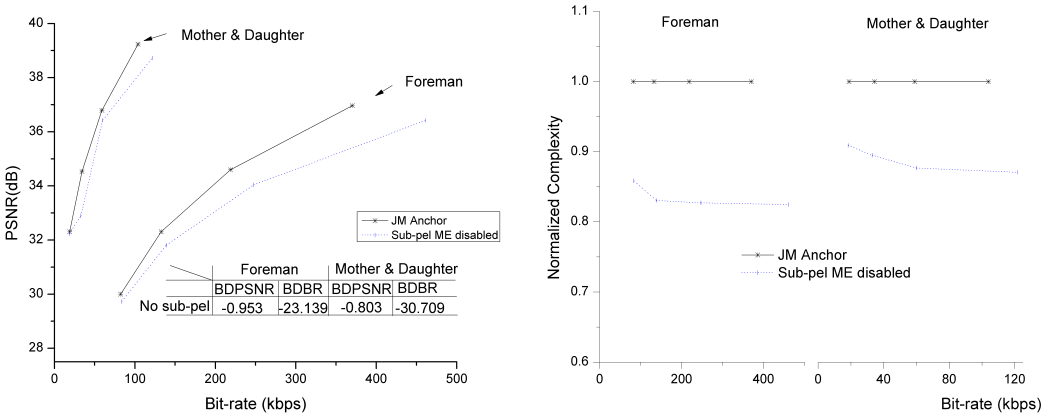
$$\begin{aligned} \underline{4} &= (\underline{1} - 5 \times \underline{2} + 20 \times \underline{3} + 20 \times \underline{5} - 5 \times \underline{6} + \underline{7})/32 \\ &= (\underline{8} - 5 \times \underline{9} + 20 \times \underline{10} + 20 \times \underline{11} - 5 \times \underline{12} + \underline{13})/32 \end{aligned} \tag{4.6}$$

Once all the half-pixel samples are available, each quarter-pixel sample is produced using bilinear interpolation between neighbouring half- or integer-pel samples as given by

$$\begin{aligned} a &= (G + \underline{3})/2 \quad \{\text{ (4) in Figure 4.4(b)}\} \\ e &= (\underline{3} + \underline{4})/2 \quad \{\text{ (5) in Figure 4.4(b).}\} \end{aligned} \tag{4.7}$$

In order to evaluate the improvement achieved using sub-pel accuracy ME, the performance of only integer-pel ME is compared with the case of considering sub-pel accuracy (denoted as JM Anchor in Figure 4.5). Figure 4.5 shows R-D and C-D performance degradation caused by not using sub-pel accuracy. Results indicate that the bit-rates are increased by 19.16% and 16.06% while decreasing normalized complexity by 10% – 18%. Therefore, sub-pel accuracy ME should

be considered in this thesis because it gives a great effect on R-D performance of encoded frames.



(a) R-D performance; R-D degradation is severe for both sequences

(b) C-D performance

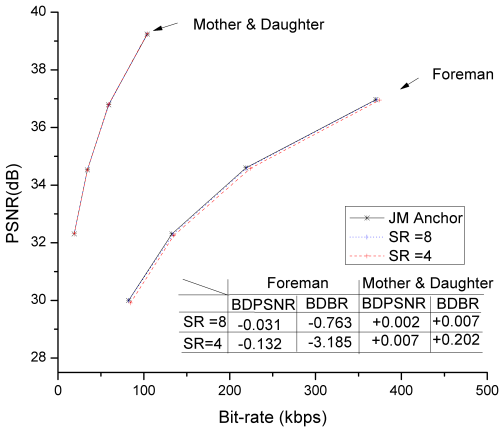
Figure 4.5: Rate-distortion performance and normalized complexity of sub-pel accuracy ME

Search Range

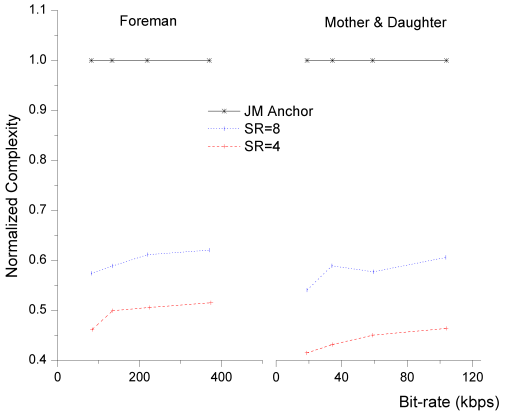
It is well known that motion search range is an important parameter in determining the coding efficiency and the encoding computational cost [52]. The most common idea to reduce search range of ME is that range is chosen on the basis of either prediction error values or of the motion vectors previously obtained for adjacent blocks. Figure 4.6 shows how much search range (SR) affects R-D and C-D performance. As search range is reduced (SR=16 $\Rightarrow$ SR=8 $\Rightarrow$ SR=4), BDPSNR is also decreased. The degradation of R-D is particularly appeared on a sequence that has active motion such as "Foreman." However, degradation of R-D is negligible in static sequences such as "Mother and Daughter."

Multiple Reference Frames

H.264/AVC allows the use of multiple reference frames, which means that the video encoder chooses among more than one previously decoded frame on which to base each macro-block in the next frame. It is common sense that the best frame for the current frame is usually the previous frame. Multiple reference frames can considerably increase encoding time because ME ordinarily carried



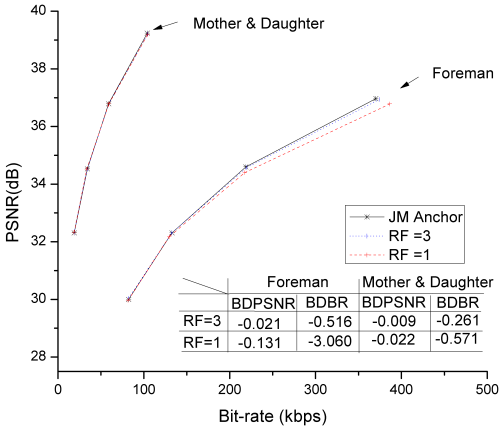
(a) R-D performance; average bit-rates are increased for Foreman by 0.6%, 2.64% at SR=8, SR=4 respectively. They are almost the same for Mother&Daughter



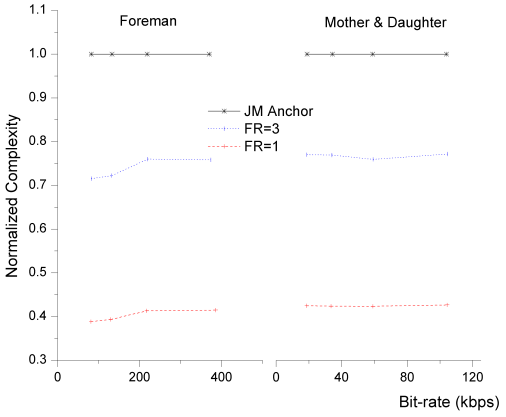
(b) C-D performance; complexity is considerably reduced according to SR

Figure 4.6: Performance variation according to search range

out only on one reference frame has to be repeated on all of the reference frames. Moreover, multiple reference frames must be stored in memory until they are no longer needed for further usage. This requires a large amount of memory usage. Based on the above two reasons, it is not feasible to consider multiple reference frames encoding on power limited especially embedded platforms. Figure 4.7 indicates that complexity is exponentially increased as the number of reference frame rises. Therefore, only single reference frame is considered in this thesis.



(a) R-D performance

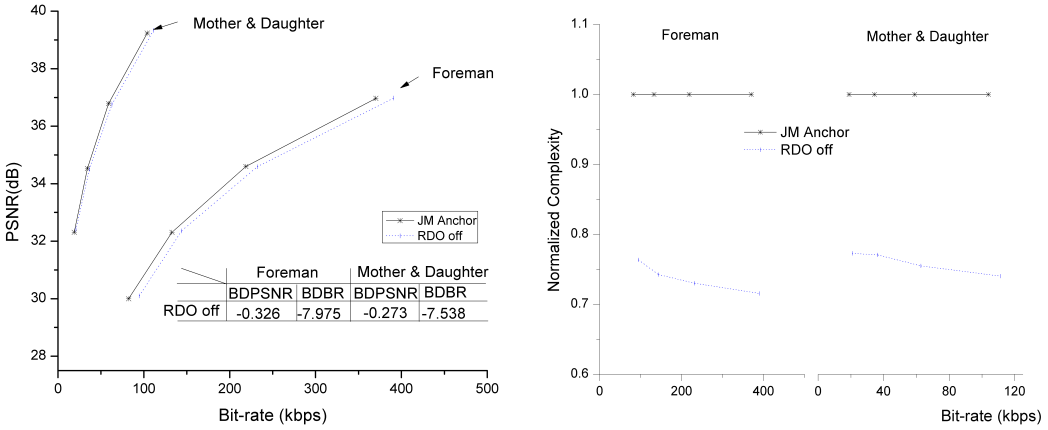


(b) C-D performance

Figure 4.7: Comparison of rate-distortion and normalized complexity according to the number of reference frames

RDO

In fact, RDO is not a part of the H.264/AVC standard, which means various RDO algorithms can be used for evaluating the performance. In RDO mode of JM, the encoder selects the best MB mode by evaluating a Lagrangian cost function for each MB. Basically, the RDO requires the MB to be encoded and decoded using all the possible modes before selecting the best mode, which increases the computational complexity. Performance is compared with the set of sequences encoded without using RDO (see Figure 4.8). Analysis reveals that the bit-rates is increased by 6.52% and 5.46% using Equation (4.3) and the computational complexity is decreased up to 30% for both sequences when RDO is not used.



(a) R-D performance

(b) C-D performance; complexity is reduced up to 30% without RDO

Figure 4.8: Rate-distortion and normalized complexity performance without RDO

Hadamard Transform

In order to understand effect of Hadamard Transform in ME, we take an example. Let all pixel values of the 4×4 reference block be 255, and those of the 4×4 current block be 0 as shown in Figure 4.9. The SAD of prediction error becomes a large value (4080). However, the SATD generates smaller value (1020) than SAD, which reduces the required bits. However, SATD is much slower than the SAD. Therefore, H.264/AVC optionally adopt SATD as a cost function in ME.

From an experiment, SATD does not give much benefits over SAD for both sequences, in terms of R-D and C-D as depicted in Figure 4.10.

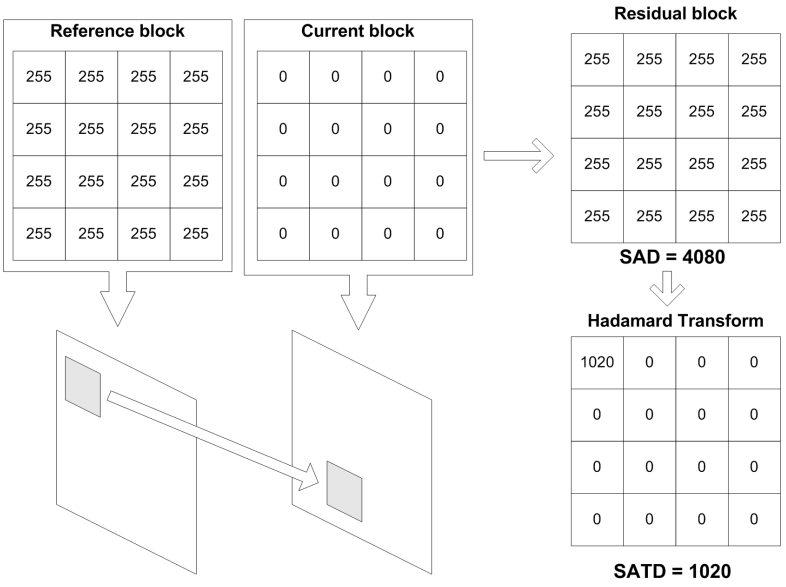
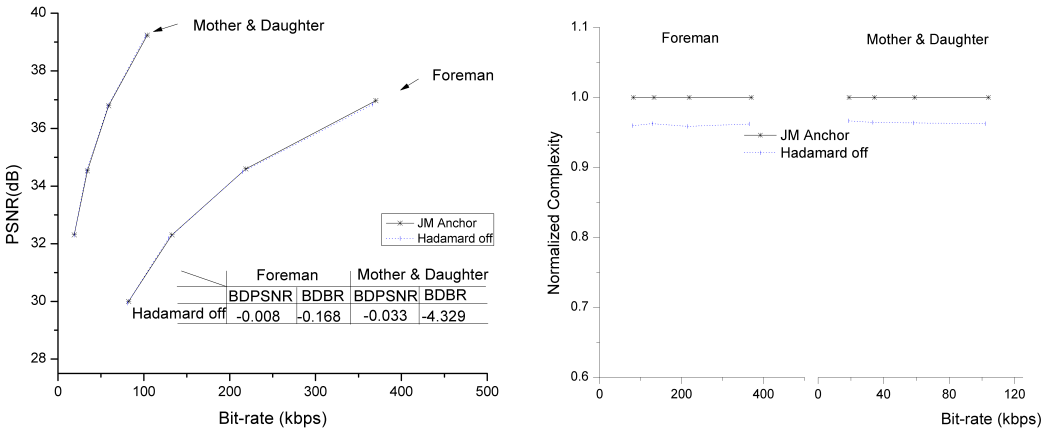


Figure 4.9: The need of Hadamard Transform



(a) R-D performance; bit-rates are increased by 0.16%, 0.66% at Foreman, Mother&Daughter respectively

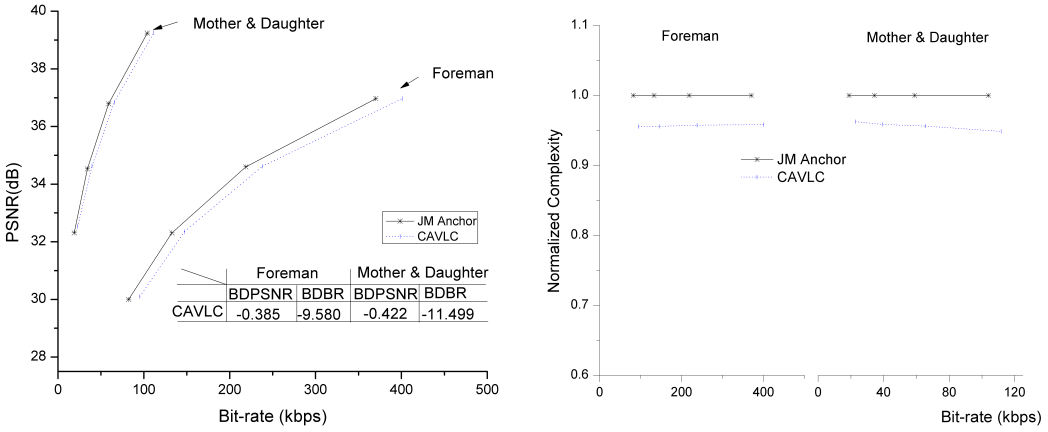
(b) C-D performance; complexity is reduced by 5% compared to JM

Figure 4.10: Rate-distortion and normalized complexity performance of using Hadamard Transform in ME

CABAC instead of CAVLC

CABAC is a form of entropy coding used in H.264/AVC. It is notable for providing considerably better R-D performance than other encoding algorithms. CABAC requires a considerable amount of processing. Therefore, CAVLC is sometimes used instead of CABAC. Figure 4.11 reveals that CABAC gives a better performance over CAVLC in terms of R-D. When CAVLC is used in the JM main profiles, the test performed in this section is performed with main profile using B-frames. the bit-rates increase by 7.7%, 8.44% for both sequences

respectively. CAVLC only is used in this thesis due to its computational efficiency.



(a) R-D performance

(b) C-D performance

Figure 4.11: R-D and C-D performance comparison with CAVLC and with CABAC (JM Anchor)

4.4 Conclusion

H.264/AVC is an international video coding standard that was jointly developed by the ITU-T and ISO/IEC. Like previous standards, H.264/AVC used block based ME. Moreover, it yields a bit-rate saving of about 50% over all previous standards as a result of its large number of encoding parameters and tools such as VBS partitioned ME, sub-pel accuracy motion vector resolution, multiple reference frames, and so on. In H.264/AVC encoders, the most computationally intensive process tends to be ME and mode decision. Therefore the main focus of recent research has been to reduce the complexity of this process.

In this chapter, encoding parameters and tools closely related to complexity are investigated with experiments using the JM reference software. From experiments, VBS , sub-pel accuracy motion vector resolution ME, and CABAC give better R-D performance. However, search range, existence of Hadamard Transform, and multiple reference frames do not give much benefit in terms of R-D and C-D performance. Based on the trade-off between R-D and C-D performance, the proposed complexity adaptation framework should consider VBS and sub-pel accuracy motion vector resolution ME because these factors

have significant influence on R-D performance. Therefore, low computational complexity version of those functions are suggested in the following chapters .

—As long as you're going to think anyway, think big.

Donald Trump

5 Low Computational Complexity Variable Block Size (VBS) Partitioning

5.1 Introduction

In recent years, the VBS ME technique has been widely employed to improve the performance of the BMA. In VBS, the block size is varied according to the type of motion. It is known to be very efficient for areas containing complex motions. However, it requires a large number of computational operations. Therefore the traditional methods to decide VBS perform it after exhaustive ME and R-D optimization. Clearly, this is not suitable for power limited platforms. Recently, there have been several attempts to reduce the computational complexity of VBS partitioning based on not performing ME in advance. In this scenario, light segmentation of a block to determine the characteristics is used, which introduces its own complexity. Therefore low complexity segmentation algorithms have become an important requirement for an encoding process. In [61], an edge block detection based subsampling method was proposed. They used Robert cross convolution masks to detect if the block was either an "Edge block" or a "Flat block". However their approach requires 15 additions and 16 absolute difference operations per 4×4 block. The approach presented in this chapter requires only 8 additions per 2×2 block. Moreover, their threshold value is decided empirically. In [51], a Cellular Nonlinear Network (CNN) type segmentation algorithm was used for detecting edge information. They used an edge enhancing low-pass filter to find regions that contain remarkable features, i.e. edges. Both prior works are performed in the pixel domain, so it is difficult

to reuse intermediate values obtained as part of segmentation. The VBS partition algorithm proposed in this chapter has two distinguishing features. Firstly, all processing is performed in the WHT domain, making it is easy to predict residue data's characteristics. Secondly, the intermediate value is reused in ME process, which is explained in the next chapter. Moreover, a computationally cost effective algorithm compared to the other related works to detect features is presented.

5.2 Walsh Hadamard Transform (WHT)

Since it is simple and efficient to execute, the WHT has been applied in many fields such as pattern matching [33], feature recognition [85, 91], wireless communication [29], and image/video compression [22, 88]. It is attractive due to the simplicity of implementation and to properties which are similar to familiar DTs. Many DTs have been used in image processing such as DCT, Discrete Fourier Transform (DFT) and Discrete Tchebichef Transform (DTT) that has recently comes under the spotlight [38, 81]. However such transforms are often hard to implement in real time in some applications due to their computational complexity of floating operations. Even when fast algorithms exist, their inverse transforms do not generate the same image as the original, which can cause a drift effect in image/video compression. In order to solve these problems, integer algorithms were developed and deployed in recent video standards such as H.264/AVC, combined with a quantisation procedure named ICT [69]. Also recently the Integer Discrete Tchebichef Transform (IDTT) was proposed in [38]. However, they focus only on 4×4 or 8×8 block sizes which are not extensible to arbitrary block sizes. Moreover, they still introduce computational complexity even though they have multiplier free structures. Although the performance of WHT is inferior to the other DTs in terms of energy compactness, it provides comparable performance on images that show less gradient changes [37]. Its computational efficiency makes it attractive in image processing to be directly applied in the transform domain since the elements of the basis kernels are orthogonal and contain only binary values (± 1). Moreover, there has been often the case that different applications require different block sizes. Therefore, efficient techniques for conversion between a block and its sub-blocks are important tasks for low complexity applications. This is mentioned in Chapter 6.

The WHT has different kinds of order at basis function; "natural order", "dyadic order", and "sequency order". The natural order of the WHT is equivalent to the post-permutation algorithm of the fast Fourier transform (FFT) and the dyadic order represents the machine-oriented algorithm of the FFT [75]. On the contrary, sequency order is analogous to frequency in DFT. The row vectors of an SWHT matrix are arranged in ascending order of sequencies which is suitable for image processing by virtue of the property as energy compaction. Thus the SWHT has been used throughout this thesis[1].

5.2.1 The Properties of the WHT

The lowest-order Walsh Hadamard Matrix (WHM) is of order two given as

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{5.1}$$

Its higher order can be obtained via a recursive method in the follow

$$H_N = \bigotimes_{i=1}^{n} H_2 = \overbrace{H_2 \otimes \ldots \otimes H_2}^{n}. \tag{5.2}$$

Let the array $[f(x,y)]$ be the intensity pixels of an original image over an array of N^2, $(N = 2^k)$, then its 2-D Hadamard Transform $[F(u,v)]$ is given as

$$[F(u,v)] = H_N[f(x,y)]H_N^T = \frac{1}{N}H_N[f(x,y)]H_N. \tag{5.3}$$

The WHT has orthogonal, symmetric, and unitary properties mentioned in Chapter 2.3 as

$$H_N H_N^T = NI, \quad H_N H_N^{-1} = NI, \quad H_N^{-1} H_N^T = NI \tag{5.4}$$

[1]Note that WHT and SWHT are the same transform in the case of 2×2 blocks

where H_N^T and H_N^{-1} represent a transpose and an inverse matrix of H_N respectively, and I is an identity matrix. Its inverse transform is expressed as

$$\begin{aligned} H_N[F(x,y)]H_N^T &= H_N H_N[f(x,y)]H_N^T H_N^T \\ &= N^2[f(x,y)], \\ [f(x,y)] &= \frac{1}{N^2} H_N[F(x,y)]H_N^T. \end{aligned} \tag{5.5}$$

The WHT has several interesting properties. The most important properties from the standpoint of image coding are dynamic range, conservation of energy, and energy compaction.

- **Bounding Dynamic range**: The DC coefficient is a measure of the average brightness of a block. If the maximum possible value of the DC is N^2A, where A is the maximum value of $f(x,y)$, the magnitude of other samples in the WHT is bounded to $\pm N^2A/2$ as mentioned in [88] and given by:

$$|F(u,v)| \leq \frac{F(0,0)}{2} \quad \text{for} (u,v) \neq (0,0). \tag{5.6}$$

- **Conservation of energy**: This is sometimes called "Parserval Theorem", which means that the power of the spatial domain is the same as that of the transform domain:

$$\sum_{x=0}^{N-1}\sum_{x=0}^{N-1} |f(x,y)|^2 = \frac{1}{N^2}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} |F(u,v)|^2. \tag{5.7}$$

- **Energy compaction**: Energy compaction capability of transforms means the capability of the transform to redistribute signal energy into a small number of transform coefficients. It can be characterized by the fraction of the total number of signal transform coefficients that carry a certain percentage of the signal energy. The lower this fraction is for a given energy percentage, the better the transform energy compaction capability. More details are mentioned in Section 5.2.2 in the case of the SWHT.

- **Convolution theorem**: It is well known for many orthogonal linear transformations that the convolution of the image is equal to the product of their transform. In the case of WHT, dyadic convolution is an analogous rule for other DTs. The detailed mathematical proof was derivered in [30].

The convolution property can be useful as a tool for image filtering in the transform domain, which is defined as follow

$$x(n) \star y(n) \stackrel{\text{def}}{=} \frac{1}{N} \sum_{k=0}^{N-1} x(k) y(n \oplus k) \iff X(n) Y(n) \tag{5.8}$$

where $X(n)$ and $Y(n)$ represent the WHT of $x(n), y(n)$ respectively, $\oplus$ is dyadic xor sum denoted as

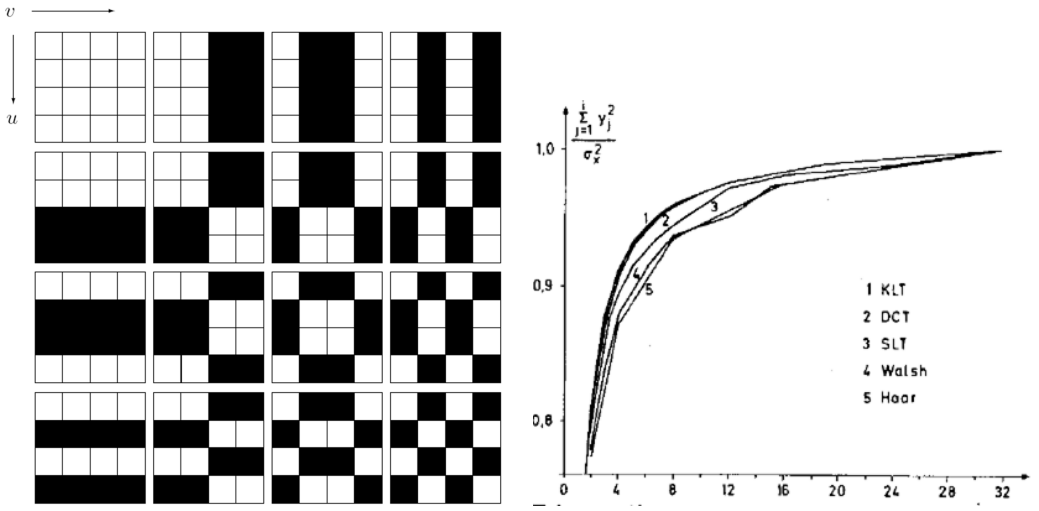
$$a \oplus b = \sum_{i=0}^{\infty} |a_i - b_i| 2^i, \text{ where } a = \sum_{i=0}^{\infty} a_i 2^i, \ b = \sum_{i=0}^{\infty} b_i 2^i. \tag{5.9}$$

5.2.2 Features of Sequency ordered Walsh Hadamard Kernels

The SWHT kernels are shown in graphical form in Figure 5.1(a) where +1 is denoted by black and -1 represented by white. The rows and columns are displayed in the ascending order of sequencies. The gain of DTs over PCM has been shown to be the ratio between the arithmetic mean and the geometric mean of the variances of all the components in the transform domain as proposed in [40], which is defined as

$$G_T = \frac{\frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \sigma^2(i,j)}{\left[\prod_{i,j=1}^{N} \sigma^2(i,j)\right]^{\frac{1}{N^2}}} \tag{5.10}$$

where N represents a block size to be transformed, $\sigma(i,j)$ is the variance of the $(i,j)^{th}$ transformed coefficient. Equation (5.10) shows the energy compactness property of DTs, which means that the large value of G_T indicates few coefficients have most of the block's energy. Figure 5.1(b) shows that the KLT is the optimal transform and the DCT performs slightly worse than KLT. The WHT shows a comparable result compared to other DTs for natural images. For example, when images are divided as 1×32 1-D arrays, only 6 out of 32 coefficients have more than 90% of the signal energy of the 1-D arrays. Therefore, it is possible to encode for 6 coefficients not for 32 coefficients with less than 10% signal loss in the case of WHT. Moreover, only 4 coefficients are needed to achieve the same result in the case of KLT and DCT. Therefore, DCT and KLT show better performance than WHT. However, the degradation of WHT over DCT and KLT is not severe when WHT performs on the frame differencing or motion compensated signal. Table 5.1 shows the G_T for three different test sequences;



(a) Sequency ordered Walsh Hadamard Transform kernel for $N = 4$

(b) Normalized transform gain of one dimensional (1x32) block for natural images [63]

Figure 5.1: The transform kernel(a) and energy compactness(b) of the WHT

(1) Foreman, (2) Mother and Daughter, and (3) Hall Monitor. It is seen that the WHT and the DCT perform very similarly for signals with low correlation such as frame differing or motion compensated frames. Therefore when the WHT is used instead of the DCT for the frame differencing image, the performance degradation of WHT over DCT is negligible.

Table 5.1: Transform gain (G_T) between DCT and WHT on frame differencing image (10th - 9th frame) divided in 16×16 block size, all sequeces are 30Hz.

	Foreman@352 × 288	*Mother and Daughter*@352 × 288	*Pedestrian*@720 × 576
DCT	*1.31*	*2.31*	*2.12*
WHT	*1.23*	*2.27*	*1.98*

5.3 Motion Edge Detection Algorithm

In this section, an algorithm for detecting motion edges using the lowest order of the WHT (2×2 block) is presented. A block with motion edges generates more inter prediction error than a homogeneous block, which is verified via mathematical analysis. After, the edge detection algorithm is presented, its results are discussed.

5.3.1 Prediction Error Analysis of Edge Gradient

Lemma 5.1. *Let Δ_d be temporal displacement of image blocks sampled at different times. The spatial gradient is denoted as $g^{'}(s)$ at location s. The prediction error σ^2 can be expressed as*

$$\sigma^2 \simeq (1 + E(\Delta_d^2)) \times (g_1^{'}(s))^2 \tag{5.11}$$

Proof. Two temporal and a spatial intensity function are defined as $f_1(x)$, $f_2(x)$ and $g_1(y)$ respectively. These are image pixels sampled at time t and $t-1$ as shown in Figure 5.2. The prediction errors are denoted as in Equation (5.12),

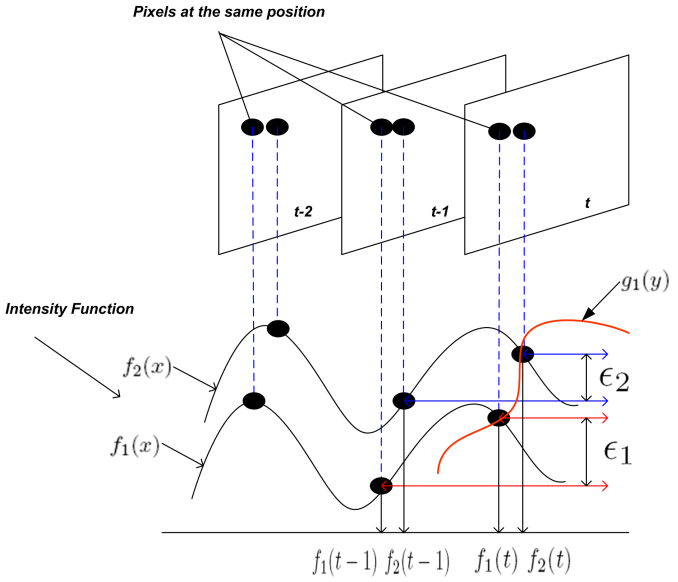


Figure 5.2: Graphical notation of terms used in the inter prediction error analysis

and its variance $(E(\epsilon_1^2)) + E(\epsilon_2^2))$ represents the total energy of the prediction errors.

$$\begin{aligned} \epsilon_1 &= f_1(t) - f_1(t-1) \\ \epsilon_2 &= f_2(t) - f_2(t-1) \end{aligned} \tag{5.12}$$

The subtraction of each prediction error is denoted in Equation (5.13), where Δ_{d1} and Δ_{d2} represent the temporal displacement errors at each temporal intensity function.

$$\begin{aligned} \epsilon_1 - \epsilon_2 &= f_1(t) - f_1(t-1) - f_2(t) + f_2(t-1) \\ &= f_1(t) - \Delta_{d1} \times f_1(t) - f_2(t) + \Delta_{d2} \times f_2(t) \end{aligned} \tag{5.13}$$

The temporal displacement errors of two blocks is the same if two pixels are involved in the same block, which is denoted as

$$\Delta_d = \Delta_{d1} = \Delta_{d2}. \tag{5.14}$$

Therefore, Equation (5.13) is rewritten as

$$\epsilon_1 - \epsilon_2 \simeq (1 - \Delta_d)(f_1(t) - f_2(t)). \tag{5.15}$$

When sampling times are sufficiently short periods, Equation (5.15) can be simplified to Equation (5.16), which has the same meaning with $g_1^{'}(y)$, at $y = s$ at time t.

$$\lim_{t \to 0}(\epsilon_1 - \epsilon_2) = (1 - \Delta_d) \times g_1^{'}(s) \tag{5.16}$$

The temporal displacement error Δ_d is a random variable with zero mean in a range of $\Delta_d \in [-search, +search]$. It is assumed that the prediction errors ϵ_1 and ϵ_2 are a memory-less stationary Gaussian source of zero means and variances (σ_1^2, σ_2^2), the total energy of prediction error (σ^2) is expressed as follows;

$$\begin{aligned}\sigma^2 = \sigma_1^2 + \sigma_2^2 = E((\epsilon_1 - \epsilon_2)^2) = E((\epsilon_1 + \epsilon_2)^2) \simeq E((1 - \Delta_d)^2) \times (g_1^{'}(s))^2 \\ \simeq \underbrace{(1 + E(\Delta_d^2))}_{\text{motion}} \times \underbrace{(g_1^{'}(s))^2}_{\text{edge}}.\end{aligned} \tag{5.17}$$

□

In Lemma 5.1, σ^2 should be linear to $(g_1^{'}(s))^2$ and $E((1 + \Delta_d)^2)$, which means that the prediction error is mainly determined by the edge gradient and motion vectors. When the current frame contains a lot of edge information, its prediction error from the previous frame will be significant. Moreover, the prediction errors are more severe if the block with edges also has motion. Therefore, VBS partitioning of those kinds of blocks should be considered to reduce the prediction errors. On the contrary, when a block is homogeneous, the redundant computational complexity can be removed without increasing prediction errors. As a result, blocks with edge information and motion (named motion edge in this chapter) need to be detected before the encoding process to reduce inter prediction error and redundant computational complexity.

5.3.2 Motion Edge Detection

From Lemma 5.1, the prediction errors of a block with motion edges are poorly estimated. VBS is the one of techniques to reduce predication errors. The recent standards such as H.264/AVC can reduce prediction errors over the previous standards to some degree by applying finite kinds of VBS. The proposed VBS partitioning has different points of view compared to H.264/AVC. In H.264/AVC, VBS is determined by selecting the block shown minimum cost after encoding with all possible block sizes. This clearly introduces computational complexity. In the proposed VBS partitioning algorithm, VBS is obtained by observing which block has motion edges. If the block has at least one motion edge, it is divided into a small size block.

The procedure of motion edge detection is as follows;

1. 2×2 blocks are selected from the top and left position of the frame, and WHT coefficients of the blocks calculated.

2. From Equation (5.7), the total energy of the 2×2 block is conserved in the transform domain. Therefore, the edge information can be obtained from the statistics of non zero sequency terms $(F(1,0), F(0,1), F(1,1))$. A good approximation of the distribution of non zero sequency terms is a variance. However, its computational complexity is too high to be applied for complexity constrained systems. Instead of obtaining the variance, the maximum values of non zero sequence terms are used to obtain similar characteristic to the variance since the dynamic range of WHT coefficients of non zero sequency term is limited by the zero sequency term as shown in Equation (5.6). When the condition of Equation (5.18) is satisfied, this block is considered as containing an edge.

$$\max_{(u,v)\neq(0,0)} F(u,v) \geq \tau \tag{5.18}$$

where τ is threshold value, which can be determined by considering how much pixel values are changed between the neighbouring pixels as an edge. τ is not particularly sensitive to the image characteristics since only a small block (2×2) is used.

3. Motion edges are obtained in the same fashion with Equation (5.18), which is slightly modified as follows;

$$\max_{(u,v)\neq(0,0)} |F(u,v)_t - F(u,v)_{t-1}| \geq \tau \tag{5.19}$$

where $F(u,v)_t$ and $F(u,v)_{t-1}$ represent the transformed block at the same location both in the current and the previous frame.

4. When a block has a motion edge, the 2×2 block is mapped to one pixel, so that a 4:1 down sampled motion edge frame (called binary motion edge map) is obtained from the original image without any further processing. It gives computational efficiency when the VBS partitioning is performed directly on the down sampled binary motion edge map.

Figure 5.3 illustrates output edge images obtained by the proposed edge and motion edge detection in comparison with Canny edge detection [12], which gives a very accurate single edge detection result. The proposed method shows more edge pixels than Canny. Canny detects a single edge line when it performs on the boundary of an object. The results of the method proposed here shows comparable edges over all image resolutions ($352 \times 288 \rightarrow 1280 \times 720$). The target of the proposed edge and motion edge detections is not obtaining a single edge (see Figure 5.3(d)) but acquiring useful regions (see Figure 5.3(b)(c)). Therefore, the proposed approach is suitable for finding remarkable region, which give a benefit for video compression. Note that edge and motion edge images are 4: 1 subsampled versions, which are intentionally displayed as the same size for clear distinction in Figure 5.3(b)(c).

Figure 5.4 shows the effect of the threshold value, τ. As τ is increased, the weak edge pixels move to background pixels. Therefore, when high QP is applied on the image, motion edge detection on the reconstructed image is equivalent to increasing τ. This is an important concept for the video encoder proposed in this thesis because we can estimate the reconstructed image without performing whole encoding process.

From a computational complexity standpoint, Canny edge detection requires several steps; 1) smoothness by applying a Gaussian filter, 2) finding gradients for each direction using Sobel or Robert operator, 3) double thresholds. A Sobel operator can be used for finding the first order gradient in Canny, but it requires 7

Figure 5.3: Motion edge and edge detection results (a) original image (all images are the 46^{th} frame); (b) 4:1 down sampled binary edge images; (c) 4:1 down sampled binary motion edge images; (d) binary edge images obtained by Canny edge detection, where double threshold values are [100, 180]

additions, 4 shift operations, 2 square operations, and 1 square root operation for every three pixels. On the contrary, the proposed method requires 8 additions for every four pixels. Table 5.2 shows the computational complexity of each method. The computational complexity of Canny edge is obtained both for a non optimized version and a highly optimized version using the IPP[2] library. The proposed approach shows faster operation compared to Canny edge (non optimized one) by a factor three. Moreover, when we apply to HD sequence, it only requires 6∼7ms per frame to detect motion edges. If we assume that the proposed method operates on a general purpose CPU, optimization is one of the

[2]Intel Integrated Performance Primitives : details are available at http://software.intel.com/en-us/intel-ipp/

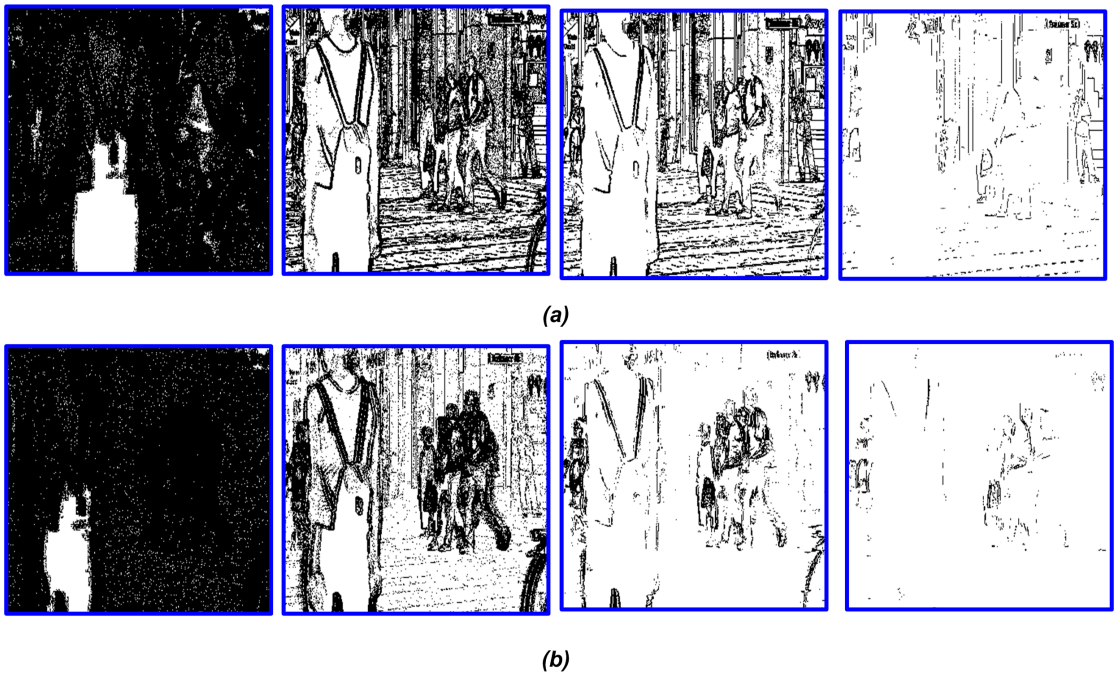


Figure 5.4: The effect of threshold value τ; (a) binary edge image (b) binary motion edge image; from the left τ is 0, 5, 10, 40 for Pedestrian sequence 46^{th} frame at 720x576

options to obtain fast operation. Sometimes it is hard on a CPU not equipped with instruction such as the Streaming SIMD Extensions (SSE). Although the proposed one shows similar performance compared to highly optimized version of Canny, it definitely shows better performance on general purpose CPU. Note that all tests are performed on an Intel^{TM} Core 2 Duo 3.0GHz with 2GB RAM using Window XP version 2002 with service pack 2 written in ANSI C++.

Table 5.2: Comparison of execution time of edge or motion edge detection; the number in () in Canny Edge denotes optimized version using IPP library.

Unit $(10 * \frac{ms}{frame})$	*Edge*	*Motion Edge*	*Canny Edge*
Mother (352x288)	*7.00*	*8.47*	*21.21 (7.54)*
Mobile(352x288)	*5.68*	*6.97*	*41.05 (15.54)*
Pedestrian(720x576)	*27.45*	*33.19*	*84.21 (28.29)*
Pedestrian(1280x720)	*63.20*	*77.02*	*152.35 (57.39)*

5.4 VBS Partitioning for ME

In this section, the VBS partitioning algorithm for ME is presented using the approach in the previous section. The relationship between the threshold value for generating a motion edge image and QP is obtained by simple mathematical analysis. Then, the VBS partitioning algorithm is explained in detail and results presented.

5.4.1 Relationship Between Threshold (τ) and QP

A memory-less Laplacian source with zero mean may provide the governing distribution for non-DC DCT or high-frequency wavelet transform coefficients [82, 96]. The characteristic of the WHT is similar to that of the other DTs. Let us suppose that the 2×2 non zero sequency WHT coefficients' residues, which are used for detecting a motion edge image as explained in Section 5.3.2, follow a zero mean Laplace distribution, i.e.,

$$p_{lap}(x) = \frac{\Lambda}{2} e^{-\Lambda|x|}, \qquad \Lambda = \frac{\sqrt{2}}{\sigma} \tag{5.20}$$

where x and σ represent the WHT non zero sequencies and their standard deviation respectively. For a given QP, the distortion is obtained as:

$$D(Q) = 2 \times \left(\int_0^{\frac{Q}{2}} x^2 p_{lap}(x) dx \right) + 2 \times \sum_{i=1}^{\infty} \left(\int_{i-\frac{Q}{2}}^{i+\frac{Q}{2}} (x - iQ) p_{lap}(x) dx \right) \tag{5.21}$$

A closed form of $D(Q)$ can be derived as

$$D(Q) = 1/2 \left(\sqrt{2} Q \mathrm{e}^{\frac{\sqrt{2}Q}{\sigma}} \left(2 - \frac{\sqrt{2}Q}{\sigma} \right) \sigma^{-1} + 2 - 2\,\mathrm{e}^{\frac{\sqrt{2}Q}{\sigma}} \right) \sigma^2 \left(1 - \mathrm{e}^{\frac{\sqrt{2}Q}{\sigma}} \right)^{-1} \tag{5.22}$$

Figure 5.5 shows the distortion against transformed coefficients' variance σ^2. For larger σ^2, the distortion is linear to Q^2. Moreover, large value of σ^2 is used for deciding edges. So, the distortion can be rewritten for a large σ as

$$D(Q) \cong kQ^2 \tag{5.23}$$

Distortion here has the same meaning as prediction errors denoted in Equation (5.17). From Equation (5.17) and Equation (5.23) , the following condition

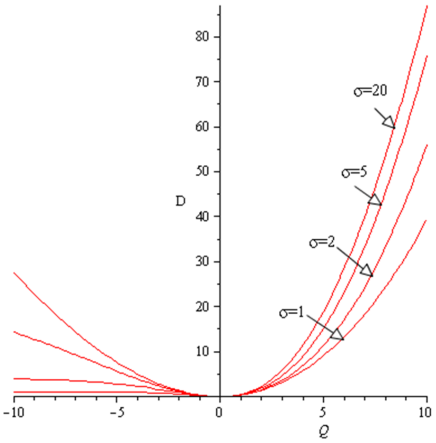


Figure 5.5: $D(Q)$ vs. variable σ; the relation $D(Q) \cong kQ^2$ is obtained for a large σ

is valid.

$$(1 + E(\Delta_{d1})^2)) \times (g_1^{'}(s))^2 \cong kQ^2. \tag{5.24}$$

Assuming that the displacement error $E(\Delta_{d1})^2$ is negligible, the edge gradient is also proportional to the threshold value, τ. Therefore the threshold value is also linear to QP:

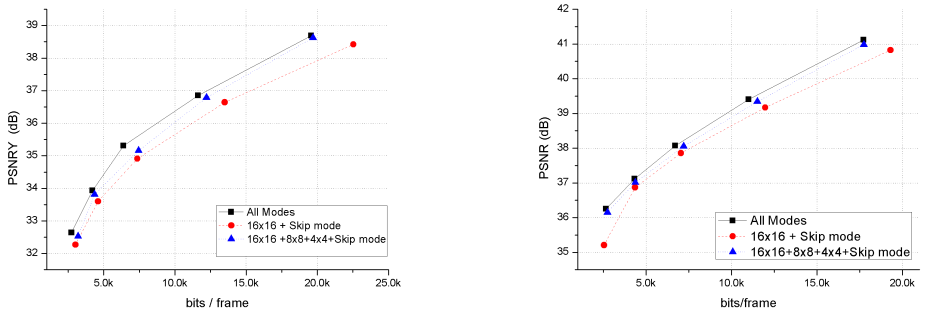
$$\tau \cong (\beta \times Q) \tag{5.25}$$

where $\beta = \sqrt{k}$.

Equation (5.25) shows that the variations of QP are similar to those of τ. The proposed VBS partitioning algorithm does not perform encoding processing; we need to estimate the output of quantised signal which is usually obtained after the encoding processing. Moreover, it is difficult to understand the behavior of QP in the pixel domain because QP works on the transformed coefficients. However, the proposed approach enables the encoder to obtain a similar image signal after QP by adjusting τ not to encode directly.

5.4.2 VBS Partitioning Algorithm

Figure 5.6 shows R-D performance by applying VBSs in JM reference software. The degradation in performance for choosing 16×16, 8×8 and 4×4 is negligible compared to using all possible modes for both slow and fast motion video. Therefore the proposed VBS partition algorithm focuses on these three kinds of block sizes.



(a) VBS performance on foreman@352x288 (b) VBS performance on mother@352x288

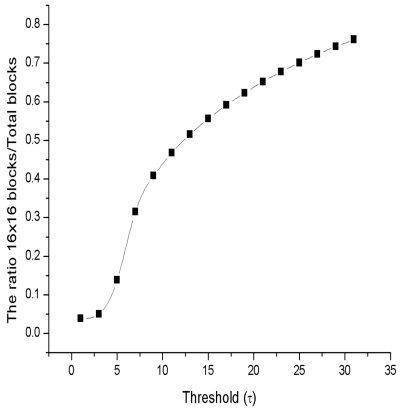
Figure 5.6: The R-D performance according to different block sizes; maximum deviation between all possible modes and 16×16, 8×8, 4×4 is less than 0.2dB

The VBS partitioning algorithm makes use of the binary motion image. The procedure is as follows;

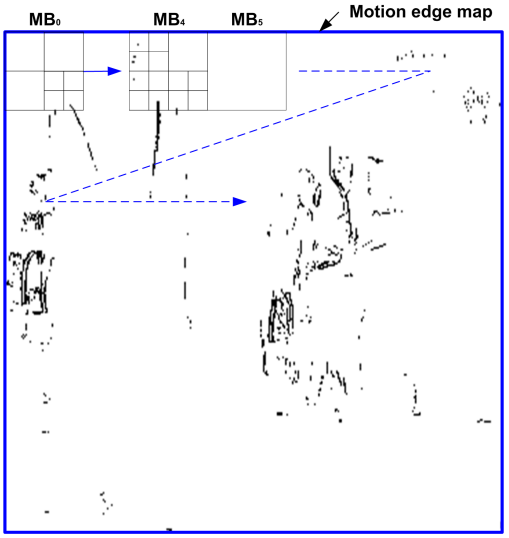
1. A 16×16 MB is chosen from the top, left position of the frame.
2. When the MB has motion edges, the block is partitioned as 8×8 or 4×4 block size depends on the location of motion edges by a recursive method.
3. Perform above procedure on the rest of MBs with raster scan order.

Figure 5.7(a) shows the ratio of deciding on 16×16 blocks out of all possible block sizes, where τ is the major criteria to decide block sizes. As τ gets large, block with weak motion edges becomes a homogeneous block. Clearly this enables us to control computational complexity automatically by not considering weak motion edges. For example, when τ is 30, the ratio of 16×16 block occupies more than 70%. The graphical illustration of VBS partitioning is depicted in Figure 5.7(b). For MB_0, the fourth quadrant is partitioned into 4×4 block size because motion edge pixels appear in that area of the MB. The other areas of the MB are partitioned into 8×8 block size. Moreover, MB_5 is partitioned into 16×16 block size because no motion edge appears in this MB.

Figure 5.8 shows the VBS partitioned image at different threshold value (τ). As τ increases, more blocks are partitioned into 16×16 block size. However, the strong motion edges still remain at high threshold value. Therefore, it is possible to decide automatically which motion edges are weak by adjusting τ.



(a) Decision of 16x16 block according to threshold value(τ); sequence is Pedestrian@720 $\times$ 576; as τ increase, 16x16 blocks are dominant as the selected block



(b) Illustration of VBS partitioning; MB_0 and MB_4 involve the sub blocks, MB_5 is considered as a 16 $\times$ 16 block on 4^{th} image in Figure 5.4(b)

Figure 5.7: Illustration and effect of the VBS partitioning

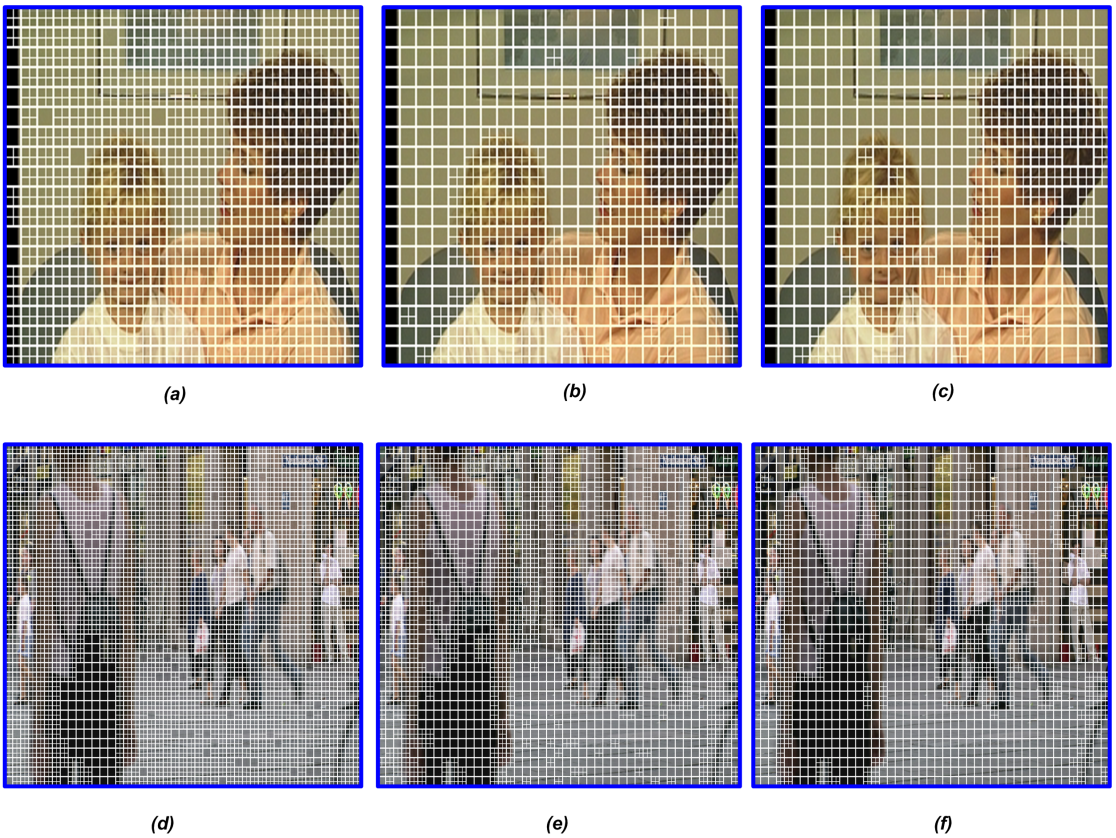
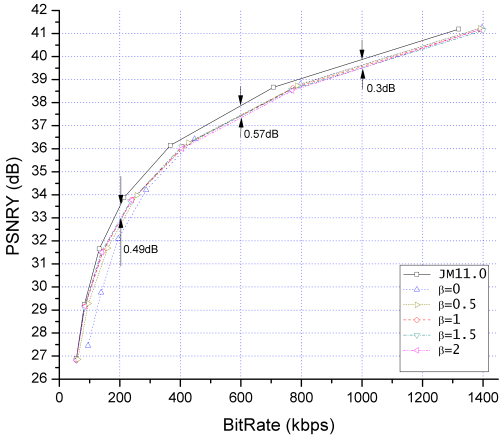


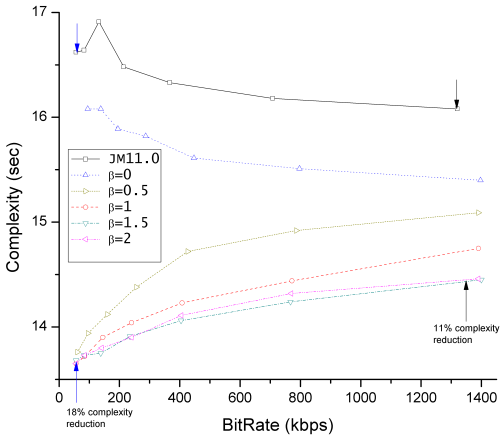
Figure 5.8: VBS partitioned results at various threshold value (τ); (a)(b)(c) are Mother sequence@352 $\times$ 288, (d)(e)(f) are Pedestrian@720 $\times$ 576; threshold value (τ) are 5, 10, 20 respectively from left side (Note that 4x4 blocks are not displayed intentionally for clear view).

5.4.3 VBS Partitioning Algorithm Performance by Choosing Optimum Threshold

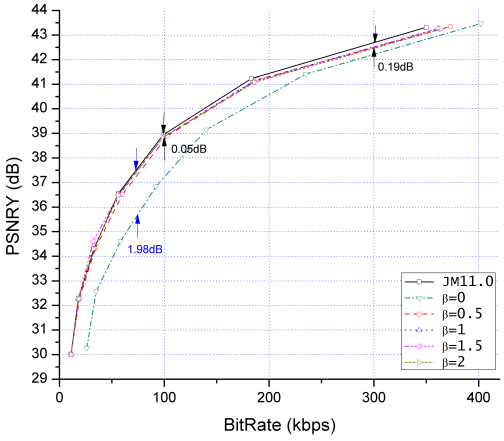
In the previous section, we showed that prediction errors are mainly affected by gradients and motion vectors as represented in Lemma 5.1. Moreover, we introduce threshold value τ, which is linearly proportional to the QP by analyzing the relationship between distortion and prediction error. The proposed VBS partitioning algorithm is integrated into the JM reference software. R-D and Complexity-Rate (C-R) performance is compared to those of JM by adjusting β at given QP as shown in Equation (5.25).



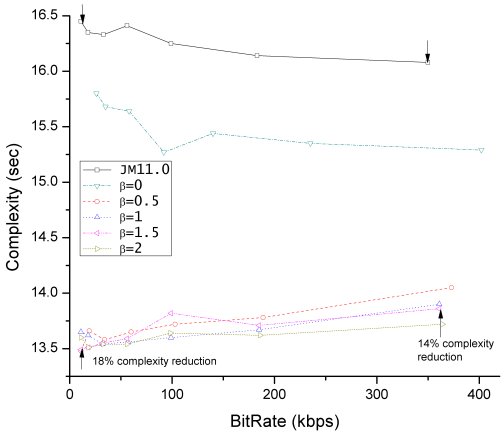
(a) R-D performance of Foreman; Maximum deviation of R-D $\beta = 1.5$ is 0.57dB at 600kbps



(b) C-D performance of Foreman; Minimum complexity reduction ratio is 11% at high bit rate



(c) R-D performance of Mother and Daughter; Maximum deviation of R-D at $\beta = 1.5$ is 0.19dB at 600kbps (black arrows), $\beta = 0$ represents 1.98dB degradation of R-D (blue arrows)



(d) C-D performance of Mother and Daughter; Minimum complexity reduction ratio is 14% at high bit rate

Figure 5.9: R-D and C-D performance for various thresholds $\tau = \beta \times QP$ at given QP; (a)&(b) Foreman@352 $\times$ 288 and (c)&(d) Mother and Daughter@352 $\times$ 288

Figure 5.9 show R-D and C-R performance at different β. From Figure 5.9(a)(c), β does not have much influence on R-D performance at high bit rates. However, β plays an important role in mid and low bit rates. For example, almost 2dB R-D degradation is shown for Mother and Daughter operating 100kbps at $\beta = 0$. Moreover, $\beta = 1.5$ shows the best performance for both sequences in R-D and C-R performance. Note that the maximum deviation of R-D performance compared to JM is less than 0.57dB in Foreman, and the minimum C-D improvement is about 11% on both sequences as shown in Figure 5.9(b)(d). We choose $\beta = 1.5$ as the optimum value for VBS partitioning.

5.5 Discussion

Despite the fact that there is a large number of VBS partitioning algorithms presented by various researches, few promising techniques can be identified as potentially useful approaches from a computational complexity perspective. In this chapter, the fundamental features of the WHT are overviewed, that is, its energy compactness, dynamic range, the conservation of energy and the convolution theorem. From Lemma 5.1, inter prediction errors are mainly caused by spatial gradients and temporal motion. Then, a computational efficient binary motion edge detection algorithm is presented in the WHT domain. The results shows that it is computational cost effective compared to the other edge detection algorithms such as Canny and Sobel operator. Moreover, the relationship between threshold value (τ) and QP is established. Finally, VBS partitioning algorithms based on motion edge detection are presented. Results show that it can be used as a basic tool for complexity adaptation in a video encoder defined in Chapter 8. A fast ME algorithm based on VBS in the SWHT domain is presented in the following chapter.

—There is no education like adversity.

Benjamin Disraeli

6 Motion Estimation based on Fast Walsh Bound Search (FWBS)

6.1 Introduction

ME is the process which generates the temporal displacements (motion vectors) that determine how each motion compensated prediction frame is created from the previous frame. A video sequence can be considered to be a discrete three-dimensional projection of the real four-dimensional continuous space-time signal. The objects in the real world may move, rotate, or deform. So the movements should be observed indirectly by projecting the light reflected from the object surfaces onto an image. However, light sources can be moved, and the reflected light varies depending on the angle between a surface and a light source. There may be objects occluding the light rays and casting shadows. Moreover, the objects may be transparent so that several independent motions could be observed at the same location of an image, or there might be fog, rain or snow blurring the observed image. In addition, the process of discretization introduces noise into the video sequence from which the video encoder estimates motions. There may also be noise in the image capture device or in the electrical transmission lines. A perfect motion model would take all those factors into account and find the motion that has the maximum likelihood from the observed video sequence. However, no such model exists. Therefore, a simple model has been applied in ME for several decades. In [90], displacement based predictive coding was presented under the assumption that the changes between successive frames are the result of the translation of moving objects in the image plane, which is called a translational model. This model is very simple; many

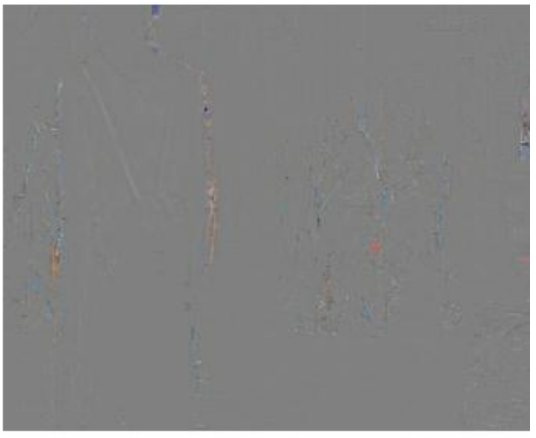
ME algorithms for 2-D images have been presented based on this simple motion model.

(a) 45th frame of Pedestrian@720x576

(b) 46th frame of Pedestrian@720x576

(c) 46th frame with overlaying motion vectors on the reconstructed image

(d) Residual error based on frame difference



(e) Residual error with motion compensation

Figure 6.1: The effect of displacement based predictive coding

Figure 6.1 shows the effect of displacement based predictive coding compared

to simple frame differencing. Figure 6.1(a)&(b) indicate the successive frames respectively, Figure 6.1(c) depicts motion vectors overlaid on the reconstructed image. Figure 6.1(d) shows a prediction error which mainly comes from moving objects boundaries. Figure 6.1(e) represents the motion compensated frame, where the prediction error is clearly dramatically decreased. In order to reduce the prediction error, ME is an essential procedure. However, ME drastically increases the computational complexity of the encoding algorithm. Therefore, many fast algorithms have been presented to reduce the heavy computational complexity burden.

In this chapter, an overview of fast motion estimation algorithms is presented. Then a fast ME algorithm named FWBS is proposed in the SWHT domain. Finally, the R-D and C-D performance of this approach is compared to those of the other well known fast ME algorithms.

6.2 Related Work

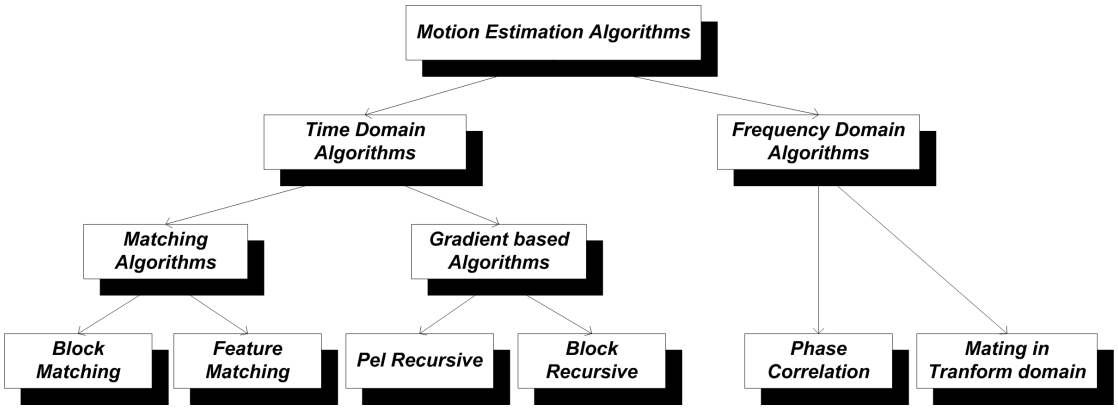


Figure 6.2: The classification of fast motion estimation algorithms [52]

Many fast algorithms have been presented in the last few decades. Clearly this is an extremely active field in the research area. The computational complexity of a ME technique can be determined by several factors such as search algorithm, search area, and cost function employed. Among them, the search algorithms play a key role in controlling of overall computational complexity and accuracy of motion. Therefore, fast ME mainly focuses on finding searching algorithms that efficiently reduce the computational complexity whilst keeping reasonable degradation of R-D performance in video compression. In Figure 6.2, the classification of ME algorithm is shown both in the time and in the transform domain

approaches suggested in [52]. More details of various fast ME algorithms are explained in the following.

6.2.1 Time Domain Algorithms

BMAs

The BMA has been widely used in video coding. The main reason for its popularity is its simplicity. It is a procedure to find a sparse motion vector field for a block. The idea behind the BMA is that the current frame is divided into non-overlapping blocks. Then, blocks in a certain search window in the previously reconstructed frame are compared to the current block and the one which leads to the best match is selected. The main difference between BMA lies in the matching criterion (cost function) and the search scheme employed to find the minimum distortion. It has been shown that the SSD indicates the best performance for ME [28]. A more comprehensive survey on BMA was given in [35]. Several famous algorithms are selected and reviewed here.

- **FS**: This algorithm is the most computationally expensive of all BMAs. It calculates the cost function at each possible location in the search window, where it finds the best possible match. It gives the lowest distortion error amongst all BMAs. In order to reduce the computational complexity of the FS, many fast BMAs have been presented, which try to achieve the same distortion error with as little computation as possible. However, the performance of fast algorithms is sometimes degraded since inappropriate initial search points generate a local minimum not a global minimum in the distortion function. Nevertheless, fast BMA is an alternative to overcome the disadvantage of the FS. Therefore fast algorithms mainly focus on reducing computational complexity without significantly sacrificing performance.

- **2-D Logarithmic Search (2D-LOG)**: Jain and Jain developed a 2D-LOG algorithm based on a 1-D logarithmic procedure [48]. Their algorithm performs under the assumption that the dissimilarity monotonically increases as the search point moves away from the point corresponding to the minimum dissimilarity. Most fast BMA algorithms are based on this assumption. The operating procedure is as follows;

1. The central point and one of the four boundaries of the search window are selected as initial searching points. Among these five points, the one corresponding to the minimum dissimilarity is picked as the winner.
2. Surrounding this winner, another set of four points are selected in a similar fashion.
3. The procedure continues until the final step, in which a set of candidate points are located within a 3×3 2-D grid.

Three Step Search (TSS) [50] and New Three Step Search (NTSS) [57] are very similar to the 2D-LOG, except these algorithms only require three steps and initial search points (nine points). In addition, many algorithms similar to 2D-LOG have been proposed, such as four step search algorithm [87], Diamond Search (DS) [107, 133], and Orthogonal Direction Search (OSA) [89]. Also the combination of these algorithms can be considered. Some of these algorithms have been adapted in the JM reference software. In [132], UMHexagonS was presented combining prediction, DS, hexagon search, partial distortion, and adaptive early termination. The approach was proven to be more robust than a single search strategy. The simplified version was also presented in [125]. In [66], the Enhanced Predictive Zonal Search (EPZS) algorithm was proposed. The EPZS defines some sets of predicted search points, which are likely to give the best match using median and temporal predictors.

- **Conjugate Direction Search (CDS)**: The CDS is another fast search algorithm presented in [59, 102]. This method can be implemented as a one-at-a-time search method parallel to one of the coordinate axes, and each variable is adjusted while the other is fixed. The procedure consists of two parts. In the first part, it finds the minimum dissimilarity along the horizontal direction with the vertical coordinate fixed at an initial position. In the second part, it finds the minimum distortion along the vertical direction with horizontal coordinate fixed position in the same fashion as the first step.

- **MBM**: To save computation in block matching, MBM was proposed using a pyramid structure, where typically a Gaussian pyramid is formed [109]. Motion search ranges are allocated among the different pyramid levels,

stating at the lowest resolution. Its computational complexity saves up to 67% without significant degradation of a reconstructed image. However, it still introduces high computational complexity compared to the other fast algorithms.

Feature Matching

In [15], edge matching from frame to frame is used for choosing the start point patterns of a search window. They used edge information to find the global minimum, called an edge-assisted search algorithm. Image features such as lines or curves are used for ME in [108]. Also background features are used in [131]. Generally speaking, feature matching algorithms are suitable for specific applications such as static scenes, video conference, and surveillance scenarios.

Gradients

The pel-recursive technique is an approach to 2-D motion estimation in image planes. Conceptually, it is a type of region matching technique. It recursively estimates motion vectors for each pixel to minimize a nonlinear function of dissimilarity between two certain regions located in two consecutive frames. There have been many enhanced versions since Netravali and Robbins published the first pel recursive algorithm [83]. A comprehensive survey of various algorithms using the pel-recursive technique can be found in [79]. Several new pel-recursive algorithms have made further improvements in terms of the convergence rate and the estimation accuracy through replacement of the fixed step size utilized in the Netravali and Robbins algorithm, which make these algorithms more adaptive to the local statistics in image frames. However, its original formulation was deterministic. The update of the ME was based on the minimization of the displaced frame difference at a pixel. It is noted that pel-recursive ME is highly sensitive to the presence of observation noise in video images. There has been much research on fast ME algorithms based on image gradient information. In [60], a Block based Gradient Descent Search (BBGDS) algorithm is presented, where the direction in which minimum is expected to lie is used to determine the search direction and the position of the next search block. In [8, 44], bidirectional gradients are used to find motion vectors. Their methods estimate the

motion between two images based on the local changes in the image intensities while assuming image smoothness, which result in comparable performance in the case of various motion models (translation, rotation, affine, and projective).

6.2.2 Frequency Domain Algorithms

Frequency-domain algorithms for motion estimation have been developed as an alternative to block matching methods. This approach has several benefits. The decoding part is no longer needed as a close-loop as in the spatial domain. It gives good performance in terms of both objective and subjective quality due to controlling higher spatial frequencies to which the HVS is less sensitive. However, it definitely introduces additional computational complexity because the recalculation of transform is performed at every shifted as part of ME.

- **DCT based approaches**: In most video standards, the feedback loop in the coder for temporal prediction consists of a DCT, an Inverse Discrete Cosine Transform (IDCT), and spatial-domain ME and motion compensation. The feedback loop limits the throughput of the coder in addition to the additional complexity attached to the overall architecture. In [49], a fully DCT based motion compensated video coder was presented, Effort was made to ensure interpretability of the pixel and the DCT domain so that DCT video codecs were fully compatible with the video coding standards. However, their algorithm introduces computational complexity for performing the DCT at every search window. Therefore, many algorithms have been presented for reducing the number of SAD in the DCT domain by introducing cost functions [72], pseudophase technique [80], and phase correlation [56].

- **FFT based approaches**: The FFT is used to obtain the frequency response to a time domain signal. The phase correlation method has been widely used for measuring motion vectors [10, 54]. It provides a very computationally efficient technique assuming a relative to large size of block. However, performance on small blocks indicates that it introduces computational complexity. Therefore, finding global motion vectors caused by camera motion is one of the promising applications of these approaches. In [45], the Windowed-Sum-Squared-Table algorithm was presented, where

the FFT was used for reducing SAD operations in the block matching. Their approaches extend a similar previous approach [26].

- **WHT based approaches**: Since the WHT consists of simple basis functions (± 1). There have been several attempts to use it in a complexity restricted application. As our proposed method also uses WHT, related work is reviewed in detail.

  Gray Code Kernels [77]: In this approach, Gray Code Kernels are used in filtering an image. The distance between a pattern and an image window is bounded by the projection value on the Walsh Hadamard basis functions. They proved the lower bound can be inferred from normalized projection kernels using a Cauchy-Schwartz inequality in the follow:

$$d_E \geq \frac{b^2}{||u||^2} \tag{6.1}$$

  where b is a projection value on vector u. They only defined a lower bound of SSD. On the contrary, our proposed method shows that lower and upper bounds on distance are restricted by its norm operations as described in Chapter 6.3.2.

  Normalized Cross Correlation (NCC) [86]: The NCC is particularly useful since it is insensitive to both signal strength and level even though NCC is computationally expensive. In [86], a computationally efficient method is proposed to generate NCC using a coarse-to-fine algorithm. As mentioned before, the phase-correlation method is suitable only for a large block size, they focused on a template matching approach using a reusable intermediate value to ensure speed up.

  Fast Walsh Search (FWS) [62] Motivated by [77], the Partial Absolute Distance (PAD) based BMA was presented named FWS in [62]. Their algorithm used block pyramid matching. For example, the zero sequency term of the $n \times n$ block is the lower bound of the distance denoted as in Equation (6.1). This block is further divided by n', and its sum of the zero sequency terms of sub blocks gives a tighter lower bound of the distance as follows.

$$d_E \geq \sum_{n'} \frac{b_{n'}^2}{||u_{n'}||^2} \cdots \geq \frac{b^2}{||u||^2} \tag{6.2}$$

As a result, they only use a zero sequency term of the WHT, which is equivalent to use the sum of pixel values of a block in the pixel domain. The FWS does not fully make use of the WHT properties such as the energy compactness of DTs. Gray Code kernel and NCC required a large block size to reduce ME error. On the contrary, FWS can be applied for small block sizes without sacrificing performance. Therefore, FWS is considered as comparison with the proposed method in terms of the WHT approach in this chapter.

6.3 Cost Functions

In order to measure the similarity between the block in the current frame and a candidate block in the reference frame, various similarity measure metrics (cost functions) are used. The SSD and the MSE are the most commonly used due to its lacks of a multiplication operation. Moreover, the bound of the SSD in the pixel domain is simply obtained by applying the L_1-norm and infinity-norm to the SWHT coefficients.

6.3.1 Similarity Measure Metrics (Cost Functions)

The goal of a ME procedure is to find the motion vector, $\vec{d} = (d_x, d_y)$ for the square current block, $C(x, y)$, so that the error between the block $C(x, y)$ and searching block in the reference frame $R(x - d_x, y - d_y)$ is minimized. Similarity measure metrics are defined as a criterion which measures the goodness of ME or how the estimation error is calculated. Several different similarity measure metrics have been proposed. Some give more robust ME and a high visual image quality, whereas some focus on reducing the computational complexity load. The computational complexity associated with some common metrics is described in the following:

- The NCC measurement between the two blocks is defined as follows [6], where the block size is $w \times h$.

$$NCC = \frac{\sum_{x=1}^{w}\sum_{y=1}^{h} C(x,y)R(x-d_x, y-d_y)}{\sqrt{\sum_{x=1}^{w}\sum_{y=1}^{h} C(x,y)^2}\sqrt{\sum_{x=1}^{w}\sum_{y=1}^{h} R(x-d_x, y-d_y)^2}} \tag{6.3}$$

  To determine motion vectors, this cost function is evaluated for selected candidate motion vectors and the maximum correlation value is chosen.

- The SSD usually yields very good performance [5], which is sometimes called the Sum of Squared Error (SSE) [14]. Its square root is also widely used as a cost function, which is called Root Sum of Squared Differences (RSSD). The MSE of an estimator is one of many ways to quantify the amount by which an estimator differs from the true value of the quantity being estimated, as well as its square root is Root Mean Squared Error (RMSE). These are all essentially minor variations of the same cost functions given as

$$\begin{aligned} SSD = SSE &= \sum_{x=1}^{w}\sum_{y=1}^{h}(C(x,y) - R(x-d_x, y-d_y))^2 = ||C-R||_2^2 \\ MSE &= \frac{1}{w \times h}\sum_{x=1}^{w}\sum_{y=1}^{h}(C(x,y) - R(x-d_x, y-d_y))^2 \\ RSSD &= \sqrt{SSD} = ||C-R||_2^1 \\ RMSE &= \sqrt{MSE}. \end{aligned} \tag{6.4}$$

  where $||\quad||_2$ is the L_2-norm.

- The SAD is widely used due to its simplicity [79]. It is one of the simplest possible metrics that takes into account every pixel in a block. However the SSD is not the best option for taking into account human perception, so the final refinement of a ME process is often done with SSD or SATD (see below). In order to compensate for degradation, a Sum of Approximate Square Difference (SASD) was presented recently in [16]. However it also introduces computational complexity compared to the SAD, even though their approach reduces the complexity of SSD by 70%. The definition of SAD is denoted as

$$SAD = \sum_{x=1}^{w}\sum_{y=1}^{h}|C(x,y) - R(x-d_x, y-d_y)| = ||C-R||_1 \tag{6.5}$$

where $||\quad||_1$ is L_1-norm.

- The SATD is widely used for block matching in ME [105, 116]. It works by taking the frequency transform of the difference between the pixels in the current block and its corresponding pixel in a search window. The SATD is less complex than the SAD, which is a critical drawback. The benefit of the SATD is that it more accurately predicts quality from both the standpoint of objective and subjective metrics. The transform used in H.264/AVC is usually the Hadamard transform for a small block (4×4 block size), where computational complexity is not negligible even though the Hadamard transform has simple basis kernels (± 1). The definition of the SATD depicted in the follows, where T represents the Hadamard transform.

$$SATD = \sum_{x=1}^{w} \sum_{y=1}^{h} |T(C(x,y) - R(x - d_x, y - d_y))| = ||T(C - R)||_1 \quad (6.6)$$

6.3.2 Bound of RSSD (SSD) in the Transform Domain

For two dimensional $\mathbb{R}^{n \times n}$ space, let $I(x,y)$ and $I^{'}(x,y)$ represent intensity functions at each pixel location (x,y) in the current and reference images. The Euclidean distance (ED) is defined in Equation (6.7) as a SSD similarity measure metric.

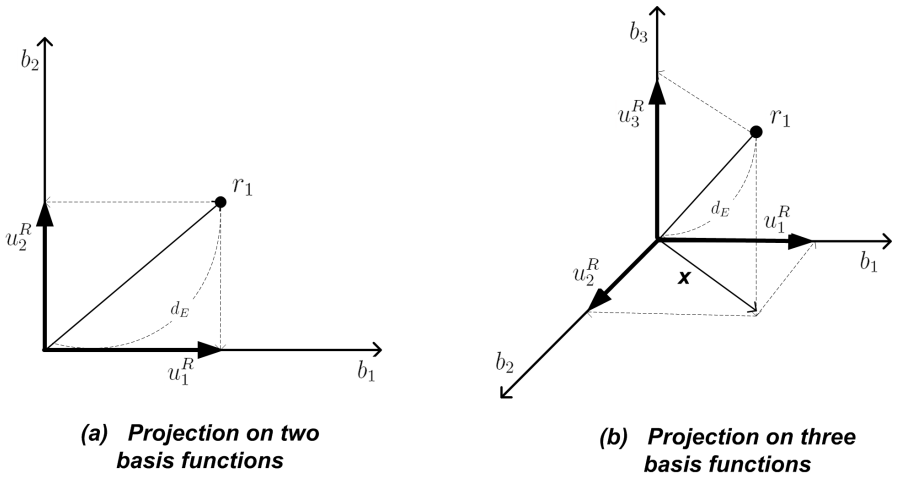
$$d_E^2(I, I^{'}) = \sum_{x,y=1}^{n} \Big(I(x,y) - I^{'}(x,y) \Big)^2 \quad (6.7)$$

Lemma 6.1. *Let $R = \{r_1, \ldots, r_{n^2}\}$ and $C = \{c_1, \ldots, c_{n^2}\}$ be the $n \times n$ 2-D vectors of the reference and the current block respectively. The transform basis functions are $B = \{b_1, \ldots, b_{n^2}\}$, and the projection vectors on B of R and C are $U^R = \{u_i^R, \ldots, u_{n^2}^R\}$ and $U^C = \{u_i^C, \ldots, u_{n^2}^C\}$ respectively. The lower and upper bounds of $d_E(R,C)$ are expressed as*

$$\left|\left|U^R - U^c\right|\right|_\infty \leq d_E(R,C) \leq \left|\left|U^R - U^c\right|\right|_1 \quad (6.8)$$

where $||\quad||_\infty$ is the infinity norm satisfying $||X||_\infty = max(|x_1|, \ldots, |x_n|)$, $X = \{x_1, \ldots, x_n\}$, and $||\quad||_1$ is a L_1-norm denoted as $||X||_1 = \sum_{i=1}^{n} |x_i|$.

Proof. To simplify this proof, Figure 6.3 is introduced to help visualize the problem. Firstly, we consider the projection of r_1 onto two basis functions of an



(a) Projection on two basis functions

(b) Projection on three basis functions

Figure 6.3: Illustration of projection of orthogonal transform onto the basis functions; Projection of $r_1 \in R$ onto (a) two basis functions and (b) three basis functions

orthogonal transform. The length from the origin to r_1, $d_E(r_1, 0)$ is defined as.

$$d_E(r_1, 0) = \sqrt{r_1^2} \tag{6.9}$$

From the triangle inequality, Equation (6.10) is obtained.

$$d_E(r_1, 0) \leq |u_1^R| + |u_2^R| \tag{6.10}$$

In a right-angled triangle, the length of an oblique side is larger than the maximum value of the other sides. Therefore, the lower bound of $d_E(r_1, 0)$ is obtained as

$$d_E(r_1, 0) \geq \max(u_1^R, u_2^R) \tag{6.11}$$

From Figure 6.3(b), it can be seen that Equation (6.11) is also valid for projection onto three basis functions. The x represents the projection vector on the b_1, b_2 plane. The lower and upper bounds are obtained as

$$\max(u_1^R, u_2^R) \leq d_E(x, 0) \leq |u_1^R| + |u_2^R| \tag{6.12}$$

From Equation (6.10), Equation (6.11), and Equation (6.12), the bounds of $d_E(r_1, 0)$ can be written as

$$\max(u_1^R, u_2^R, u_3^R) \leq d_E(r_1, 0) \leq |u_1^R| + |u_2^R| + |u_3^R| \tag{6.13}$$

When r_1 is extended to vector area R and the projection on basis functions U^R, the following is observed by recursion of the operations above.

$$\left|\left|U^R\right|\right|_\infty = \max(u_1^R, \ldots, u_n^R) \leq d_E(R, 0) \leq \sum_{i=1}^{n^2} |u_i^R| = \left|\left|U^R\right|\right|_1 \tag{6.14}$$

Finally, if two vectors R and C are considered, Equation (6.14) is extended to the difference of two vectors.

$$\begin{aligned}\left|\left|U^R - U^C\right|\right|_\infty &= \max(u_1^R - u_1^C, \ldots, u_n^R - u_n^C) \\ &\leq d_E(R, C) \leq \sum_{i=1}^{n^2} |u_i^R - u_i^C| = \left|\left|U^R - U^C\right|\right|_1\end{aligned} \tag{6.15}$$

□

From Lemma 6.1, the lower bound of the SSD between two blocks' pixel values is restricted by the maximum value of transformed coefficient, and the upper bound is also restricted by a L_1-norm of the transformed coefficients. When transforms such as the DCT or the SWHT are used for calculating the SSD, the DC or zero sequency term has a high probability to be a maximum number. If the transformed coefficients are ordered in decreasing energy, such as a zigzag scan, the SSD of the pixel domain can be obtained by calculating the most significant coefficient's SAD operations. Therefore, a small number of operations give a good performance and low computational complexity according due to the energy compactness characteristic of orthogonal transforms. For example, when we choose two blocks, A, B, whose pixel values are the same, i.e. $||A - B||_\infty = ||A - B||_1$, then $d_E(A, B)$ is the same as its infinity-norm. Therefore, we can easily calculate the RSSD only taking the absolute value of the DC terms of the two blocks in the transform domain. This is a special case of Lemma 6.1. This provides motivation for fast ME using SWHT in this chapter.

6.4 The FWBS Algorithm for Motion Estimation

In this section, we propose the *FWBS* algorithm that performs block matching in the SWHT domain and provide its performance results. Using Lemma 6.1, massive reduction of the SAD operations is achieved by adapting it to work in the transform domain.

6.4.1 Fast Sequency ordered Walsh Hadamard Transform

From Lemma 6.1, the RSSD is restricted from infinite-norm to L_1-norm of the transformed coefficients. The SWHT has an energy compactness property like other orthogonal transforms explained in Section 5.2.1. Therefore, when SWHT coefficients of a 16×16 block are used, the zero sequency term might be the value of the infinity-norm. Summing the rest of the coefficients to the zero sequency term makes the upper bound be close to the RSSD. 16 lower frequency coefficients out of 256 coefficients are enough to obtain the upper bound of the RSSD, which is the main way to reduce the number of SAD operations. However, the SWHT of a whole block is required, which leads to high computational complexity even though only 16 coefficients of a block are needed. Therefore, a computational cost effective algorithm for the SWHT is a crucial part of the fast ME proposed here. A fast SWHT method is presented by utilizing the relationship between a block and its sub-blocks in this section.

Lemma 6.2. *Given a block of $n \times n$ X_{nn}, where $n = 2^k$, divided into its sub blocks $\frac{n}{2} \times \frac{n}{2}$, four sub blocks' SWHT are denoted as X_1, X_2, X_3 and X_4. The relationship of the SWHT between a block and its sub blocks can be obtained as*

$$X_{nn} = (I_{\frac{n}{2^3}} \otimes S_4) SWHT_{2\times 2}(Q_{nn}) \tag{6.16}$$

where $\otimes$ is Kronecker multiplication, $SWHT_{2\times 2}$ is a 2×2 SWHT, I and Q_{nn} are a identity matrix and reordered sub blocks' SWHT coefficients respectively. Q_{nn} is denoted as:

$$Q_{nn} = \{X_1(0), X_2(0), X_3(0), X_4(0), \ldots, X_1(\frac{n}{2}-1), X_2(\frac{n}{2}-1), X_3(\frac{n}{2}-1), X_4(\frac{n}{2}-1)\} \tag{6.17}$$

and S_4 is a reordering diagonal matrix defined as:

$$S_4 = \begin{bmatrix} I_2 & 0 & 0 & 0 \\ 0 & B_2 & 0 & 0 \\ 0 & 0 & \overline{I_2} & 0 \\ 0 & 0 & 0 & B_2\overline{I} \end{bmatrix} \tag{6.18}$$

where B_2 *and* $\overline{I_2}$ *are vertical transition matrix and reverse identify matrix denoted as*

$$\begin{bmatrix} x_2 & x_1 \\ x_4 & x_3 \end{bmatrix} = B_2 \begin{bmatrix} x_1 & x_2 \\ x_3 & x_4 \end{bmatrix}, \quad \overline{I_2} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \tag{6.19}$$

Proof. Detailed proof is presented in Appendix A. □

Lemma 6.2 indicates that the transformed coefficients of a $n \times n$ block could be obtained via its sub partitioned four $\frac{n}{2} \times \frac{n}{2}$ SWHT coefficients without an inverse transform. It gives a hint to calculate the required coefficients separately since all the coefficients can be calculated by 2×2 order independently, which gives additional benefit for reducing computational complexity. Moreover, 2×2 SWHTs of an image are already calculated as part of the VBS partitioning procedure in Chapter 5. Therefore, the best option is that these values are reused in the ME process.

Assume that the lower 4×4 coefficients of a 16×16 block are needed to find motion vectors in the search window. Figure 6.4 visualizes this assumption. Let the search range be p, then the size search window becomes $2p + 1$ in the reference frame as shown in Figure 6.4(a). Assuming a 16×16 block is the required size for a matched block. The procedures of obtaining the lower 4×4 coefficients are as follows:

1. The searching window of 16×16 in the reference frame is equally divided into sixteen 4×4 sub blocks as shown in Figure 6.4(b), and zero sequency terms of each sub block are calculated.

2. Using four zero sequency terms in the first quadrant, 2×2 SWHT on those 4 coefficients makes 4 lower coefficients of a 8×8 block as shown in Figure 6.4(c). The same procedures are performed in the other quadrants.

3. Using a reordering matrix mentioned in Lemma 6.2, the lower 4×4 SWHT coefficients in a 16×16 block are calculated as depicted in Figure 6.4(d).

4. Scan the coefficients in zigzag order, which means that coefficients are considered in order its energy.

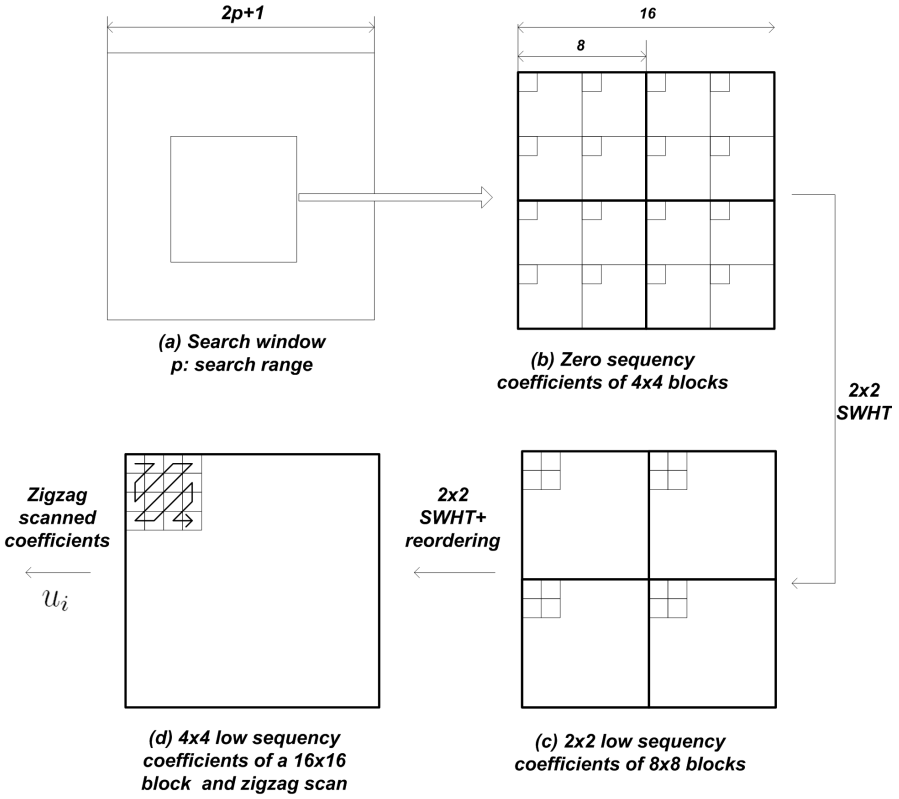


Figure 6.4: The procedure for obtaining lower 4×4 coefficients in a 16×16 block using sub-blocks' coefficients using Lemma 6.2

For illustration, we compare the proposed fast SWHT with the fast algorithms presented in [70, 71]. These existing fast algorithms the number of additions for a $n \times n$ block transform is

$$C_{exist} = 2n^2 \log_2(n). \tag{6.20}$$

On the contrary, the proposed method requires

$$C_{propose} = \overbrace{n^2}^{\text{\# of additions for 4x4 zero sequency terms}} + \overbrace{8^2}^{\text{\# of 2x2 SWHT}}. \tag{6.21}$$

For example, the proposed fast SWHT for a 16×16 block shows a complexity gain by $\frac{2048}{320} = 6.7$ times compared to existing fast algorithms.

6.4.2 The Proposed Fast Motion Estimation Algorithms

An illustration of the FWBS is depicted in Figure 6.5(a). ME begins with the origin point $(0,0)$, which could be the same or the prediction position of the

current block in the reference frame. The procedure is as follows:

1. Let the SWHT coefficients of the search window matched with the current block be u_i^R, the coefficients of the current block be u_i^C. The initial distortion d_{index} is obtained as in Equation (6.22), and saved as an index.

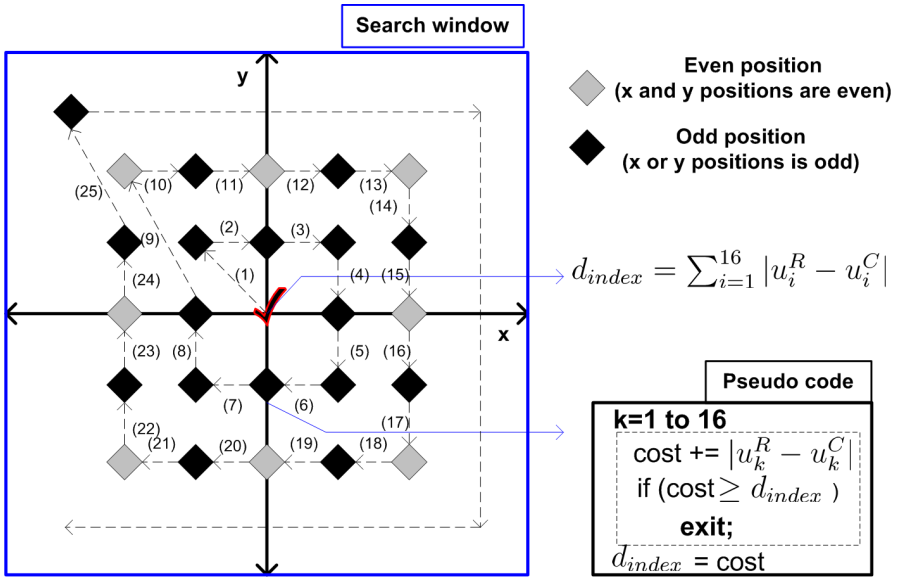
$$d_{index} = \sum_{i=1}^{16} |u_i^R - u_i^C| \tag{6.22}$$

2. Using spiral search patterns (see numbers of Figure 6.5(a)), the distortion for the next searching point is compared with an index. If the distortion is larger than an index in the middle of performing Equation (6.22), no more calculation is needed, and the routine is escaped. For some blocks with little correlation a few sequency terms are needed instead of checking all 16 coefficients, which leads to computational complexity saving. Then the search point moves to the next position, and the same processing is performed in all searching positions.

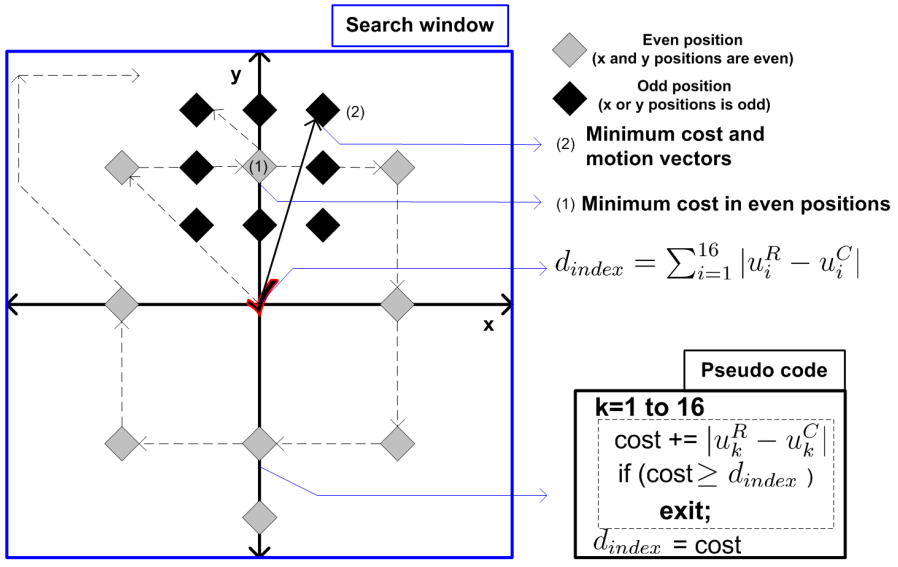
3. Find the global minimum distortion in all search points, then save motion vectors for a given block.

Note that the SWHT coefficients obtained as part of the VBS partitioning could be reused in the ME denoted as the even position in Figure 6.5(a).

To further reduce computational complexity, the FWBS is performed only on even positions to find the minimum cost as shown in Figure 6.5(b), called Fast sequency ordered Walsh hadamard transform Bounding Search for Reusable block (FWBSR) . It is a complexity efficient method because the SWHT coefficients at even positions are already obtained in the middle of performing VBS. Therefore, no further computational complexity is introduced except final refinement of integer-pel motion vectors. This approach has the same meaning as performing integer-pel ME on a 2:1 down scaled image. The procedure consists of two steps. In the first step, the minimum cost is obtained and saved as an index after searching all possible even positions in the search window. At the next step, the minimum costs of eight integer-pels near the index position are calculated and compared with each other. The position that indicates the minimum is taken as the motion vector.



(a) FWBS spiral search pattern and procedure for ME; Start from origin point, and find the minimum cost of all positions



(b) FWBSR spiral search pattern and procedure for ME; Start from origin point. First, find the minimum cost of all even positions. Second, find the global minimum of 9 points (8 positions + minimum cost position)

Figure 6.5: The search pattern and procedure for FWBS and FWBSR. k represents 16 coefficients from DC

6.4.3 Results of Motion Estimation

Computational Complexity Analysis

The computational complexity of the FWBS is compared to the FS. Equation (6.23) shows the computational complexity of the FWBS. The first term indicates the number of additions to obtain 16 coefficients of the SWHT as shown in Equation (6.21). The second term represents the number of searching points $(2p+1)^2$. The proposed method requires sub partitioned block size up to $(\frac{n}{2^2})^2$ for a given n^2 block . When n is less than 8, no sub block partition is available, thus fast algorithms are used for obtaining a block's coefficients, where the number of additions is $2n^2 \log_2 n$.

$$N_{FWBS+} = \begin{cases} (n^2+64)((2p+1)^2+1) & \text{for } n \geq 8\,, \\ (2n^2 \log_2 n)((2p+1)^2+1) & \text{for } n < 8. \end{cases} \tag{6.23}$$

In the case of the FWBSR as shown in Equation (6.24), the number of additions is only required at even positions and final refinement of 8 integer-pels. Moreover, it can be reduced by reusing pre-calculated coefficients as part of the VBS partitioning. Therefore, the FWBSR gives computational complexity benefits both in terms of the number of calculating SWHT coefficients and the searching points.

$$N_{FWBSR+} = \begin{cases} (n^2+64)(\frac{(2p+1)^2}{4}+9) & \text{for } n \geq 8\,, \\ (2n^2 \log_2 n)(\frac{(2p+1)^2}{4}+9) & \text{for } n < 8. \end{cases} \tag{6.24}$$

In terms of SAD operations, the FWBS and FWBSR performs an early termination if the current summing SAD is larger than the index distortion d_{index} in Equation (6.25). For example, "κ is one" means the first SAD is larger than d_{index}, and no further processing is needed. The maximum value of κ is 15.

$$N_{SAD} = \begin{cases} E[(16-\kappa)] \times (2p+1)^2 \leq 16(2p+1)^2, & \text{for FWBS} \\ E[(16-\kappa)] \times (\frac{(2p+1)^2}{4}+8) \leq 16(\frac{(2p+1)^2}{4}+8), & \text{for FWBSR} \end{cases} \tag{6.25}$$

where $E[x]$ represents a expectation of x. For the FS, the total number of SAD is denoted as:

$$N_{SAD} = n^2(2p+1)^2 \tag{6.26}$$

In comparison with the FS, we should consider the complexity of the SAD operation. The FWBS focuses on reducing the number of SAD operations. Before analyzing the computational complexity, we note the difference between a SAD operation and an addition. To perform a SAD operation on a general purpose processor without specific mnemonics such as the SAD, more CPU clock cycles are need compared to an addition or a subtraction which requires only one cycle in most common CPU. The pseudo code of a SAD is depicted as follows;

```
1: move  r, a
2: cmp  r, 0
3:    jge     jump
4:    neg  r
jump:
5: mov return, r
```

Table 6.1: Equivalent complexity comparison between FWBS and FS for a block; () indicates the complexity gain of the FWBSR, sr represents search range

	sr	FS	FWBS	FWBSR	Complexity gain
n=16	4	103,680	6,800	2,580	15.25(40.19)
	8	369,920	23,440	6,740	15.78(54.88)
	16	***1,393,920***	***87,440***	***22,740***	***15.94(61.23)***
n=8	4	25,920	6,608	2,388	3.92(10.85)
	8	92,480	23,248	6,548	3.98(14.00)
	16	***348,480***	***87,248***	***22,548***	***3.99(15.46)***
n=4	4	6,480	*6,544*	2,324	-0.01(2.79)
	8	23,120	23,184	6,484	-0.01(3.57)
	16	***87,120***	***87,184***	***22,484***	***0(3.87)***

To implement a SAD operation on a general purpose CPU, four CPU cycles are needed at least. This shows that the maximum complexity cost of a SAD, C_{SAD}, is almost five times of that of an addition or subtraction. Thus, it is unfair comparison if the computational complexity of the two methods is considered the same. We assume that the conversion relationship between a SAD and an addition in terms of computational complexity is a factor of five as in Equation (6.27).

$$C_{SAD} \simeq 5C_{addition} \tag{6.27}$$

Table 6.1 shows the equivalent computational complexity. The FWBS and FWBSR show high complexity gain. For example, the FWBS and FWBSR are faster than the FS by a factor of 16 and 61 respectively. As the block size is decreased, the complexity gain is also decreased. Search range does not significantly affect the complexity gain at a fixed block size. When the FWBS is applied to a large block, its complexity gain increases exponentially compared to the FS. Template matching is a good candidate application of the FWBS since it requires a large block.

Comparison with the Other Fast Motion Estimation Algorithms

We simulate the performance of the FWBS under the condition that the block size is chosen as 16×16 pixels, the maximum displacement is ± 15, and the accuracy is integer-pixel. The test video sequences used are the "Foreman", "Mother and Daughter", "Pedestrian", "Rush Hour", and "Blue Sky". Their characteristics are described in detail at Chapter 3. Objective criteria MSE are applied to measure the quality of ME. Three different well known fast algorithms are chosen to be compared to the FWBS, these are TSS, NTSS, and DS. In addition, the FWS algorithm is also compared to the FWBS because they both use the WHT. The main difference between the FWBS and fast ME algorithms is controlling not the number of searching points but the number of SAD operations. It turns out that the FWS gives the best accuracy as shown in Table 6.2. In particular, it does not require a full WHT since Pseudo Sum of Absolute Difference (PSAD) could be obtained using only zero sequency terms of separated blocks, which are calculated just by summing up all pixels value in the blocks. All fast algorithms are compared to the FS in terms of the MSE degradation, which is also redefined as "PSNR drop", and execution time, named "Time Saving". The FWBS provides quite accurate and reliable ME performance for most video sequences from QCIF to HD and 30% faster compared to the other fast algorithms performed in the pixel domain. The PSNR gain of the FWBS over the other fast algorithms is shown from 0.9dB at "Pedestrian" to 0.1dB at "Mother and Daughter". Clearly this is not the case for FWBSR that shows drop off greater than 2dB for "Pedestrian" and "Rush hour". However, its computational complexity saving might be a good attractive feature in some applications.

Table 6.2: Comparison with other fast ME algorithms; FWBS shows a comparable performance both a search time and MSE error; yellow highlight represents the result of the proposed method

Sequences		FPS	Search Algorithm	MSE (a) per a frame	Search Time (ms) per a frame (b)	PSNR drop (dB) 10log((a)FS/(a))	Time saving (%) ((b)FS -(b))/(b)FS
QCIF 176x144	Foreman	30	FS	963.85	406	0.00	0
			TSS	1,118.23	18	+0.64	95.5
			NTSS	1,210.12	19	+0.99	95.3
			DS	1112.21	18	+0.62	95.5
			FWS	987.20	102	+0.10	74.9
			FWBS	1,081.43	14	+0.49	96.4
			FWBSR	1,496.84	5	+1.90	98.6
	Mother & daughter	30	FS	88.28	406	0.00	0
			TSS	89.39	18	+0.05	95.5
			NTSS	90.21	18	+0.09	95.5
			DS	92.12	20	+0.18	95.1
			FWS	88.60	108	+0.02	73.4
			FWBS	89.95	14	+0.08	96.6
			FWBSR	97.25	5	+0.4	98.6
CIF 352x288	Foreman	30	FS	814.64	1,636	0.00	0
			TSS	1,104.15	72	+1.32	95.6
			NTSS	1,098.21	74	+1.30	95.5
			DS	1,124.32	69	+1.40	95.8
			FWS	868.32	413	+0.27	74.6
			FWBS	1,010.84	58	+0.94	96.4
			FWBSR	1,401.65	22	+2.35	98.6
	Mother and daughter	30	FS	119.43	1,656	0.00	0
			TSS	131.00	77	+0.40	95.0
			NTSS	127.81	79	+0.29	95.2
			DS	133.98	78	+0.50	95.3
			FWS	121.21	412	+0.06	75.1
			FWBS	127.87	58	+0.29	96.5
			FWBSR	166.05	22	+1.43	98.6
SD 720x576	Pedestrian	25	FS	1,262.18	6,476	0.00	0
			TSS	1,767.11	284	+1.46	95.6
			NTSS	1801,11	277	+1.54	95.7
			DS	1782.32	289	+1.50	95.5
			FWS	1310.21	1,345	+0.16	79.2
			FWBS	1,450.15	239	+0.60	96.3
			FWBSR	2,373.31	91	+2.74	98.6
	Rush Hour	25	FS	606.70	6,573	0.00	0
			TSS	713.56	290	+0.70	95.6
			NTSS	708.23	287	+0.67	95.6
			DS	720.31	288	+0.75	95.6
			FWS	630.21	1,352	+0.17	79.1
			FWBS	698.18	238	+0.61	96.4
			FWBSR	1022.61	91	+2.26	98.6
HD 1280 x720	Pedestrian	25	FS	620.10	14,062	0.00	0
			TSS	691.78	627	+0.48	95.5
			NTSS	701.11	630	+0.53	95.5
			DS	698.31	645	+0.52	95.4
			FWS	640.78	3,321	+0.14	76.4
			FWBS	670.21	528	+0.34	96.2
			FWBSR	1,022.61	199	+2.17	98.6
	Rush Hour	25	FS	251.89	14,376	0.00	0
			TSS	358.58	636	+1.48	95.6
			NTSS	363.21	628	+1.59	95.6
			DS	360.98	632	+1.56	95.6
			FWS	267.21	3,210	+0.26	77.7
			FWBS	297.87	527	+0.73	96.3
			FWBSR	465.18	201	+2.67	98.6

Table 6.3: The performance of the fast algorithms for camera rotation; the fast algorithms NTSS and TSS do not find motion vectors properly, on the contrary, FS, FWS, FWBS, FWBS, and DS shows comparable results.

Sequences	FPS	Search Algorithm	MSE (a) per a frame	Search Time (ms) per a frame (b)	PSNR drop (dB) 10log((a)FS/(a))	Time saving (%) ((b)FS -(b))/(b)FS
Blue Sky @720x576	25	FS	823.10	6621	0.00	0
		TSS	7,027.00	288	+9.31	95.7
		NTSS	8,232.21	296	+10.00	95.5
		DS	890	285	+0.34	95.7
		FWS	852.32	1,318	+0.15	80.0
		FWBS	888.40	231	+0.33	96.5
		FWBSR	1021.21	92	+0.94	98.6
Blue Sky @1280x720	25	FS	358.61	13,928	0.00	0
		TSS	4,527.08	654	+6.29	95.3
		NTSS	4,629.65	633	+6.57	95.5
		DS	412.21	645	+0.61	95.4
		FWS	369.21	3,159	+0.13	77.3
		FWBS	388.51	517	+0.35	96.3
		FWBSR	452.23	198	+1.01	98.6

The TSS, the NTSS, and the DS show comparable performance. The reduction in the number of search points is usually based on the assumption that the MSE increases monotonically as the search point moves away from the global minimum. However, in the "Blue Sky" sequence, which contains camera rotation, the performance of TSS or NTSS shows a significant degradation. TSS has been the most popular approach due to its simplicity. However, TSS suffers from two problems: first, its PSNR is substantially lower than that of FS and second, it can be easily trapped in a non-optimum solution especially not in translate motion model. Table 6.3 show the performance of the TSS working on "Blue Sky" sequences. The performance is degraded by almost 10dB, it is hard to adapt TSS in these kinds of video sequences. However, the FWBS, FWBSR and the FWS show good performance on these video sequences, since these algorithms focuses not on reducing the number of searching points but eliminating the number of SAD operations. Moreover, DS shows a comparable result. Therefore, search patterns play an important role in ME for those sequences.

Performance comparison with Fast MEs in the JM reference software

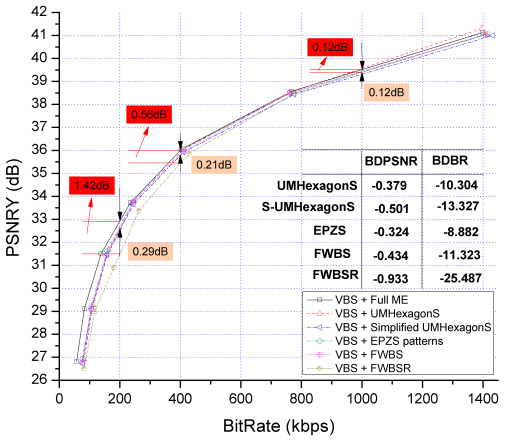
ME for H.264/AVC enables sub-pel accuracy motion vectors up to quarter-pel, and two separated stages for ME was adapted in the JM. Firstly, integer-pel ME is finding a best matched integer pixel position in the reference frames. Following

the integer-pel ME, sub-pel ME is performed around the best integer position to further reduce the prediction error. The search range of sub-pel ME in JM is limited not to reach other neighbouring integer-pel positions.

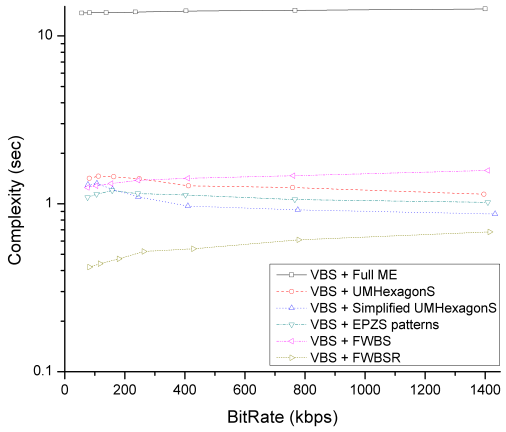
The FWBS is integrated into JM by replacing integer-pel ME of the JM, which is compared to the other fast ME. Several fast ME algorithms were integrated into JM reference software to reduce the encoding time of the ME process such as UMHexagonS [132], Simplified UMHexagonS [125], and EPZS pattern [66]. These algorithms show that more than 90% of ME time can be removed from FS while very good R-D performance is still maintained. The FWBS is compared to these algorithms in terms of R-D and C-R performance. We test FWBS algorithm on Foreman and Mother and Daughter sequences. For the FWBS, the PSNR drop is less than 0.29dB at 200kbps, which is the same as the PSNR drop obtained by the other fast ME algorithm as shown in Figure 6.6(a). If the FWBSR is used instead of the FWBS, PSNR drop is not severe or at least the same as other fast algorithms at high bit rate (low QP). On the contrary, a considerable PSNR drop occurs when high QP (low bit rate) is applied, 1.42dB PSNR drop compared to the JM. Turning to C-R performance, the FWBS shows slightly higher complexity compared to the other fast algorithms, but it is negligible. More computational complexity saving can be achieved when the FWBSR is used. In a static sequences such as Mother and Daughter, the FWBS and FWBSR show the almost same R-D performance in comparison with the other fast MEs as shown in Figure 6.6(c). Therefore, FWBSR can be used for static sequences without major degradation of the R-D.

6.5 Discussion

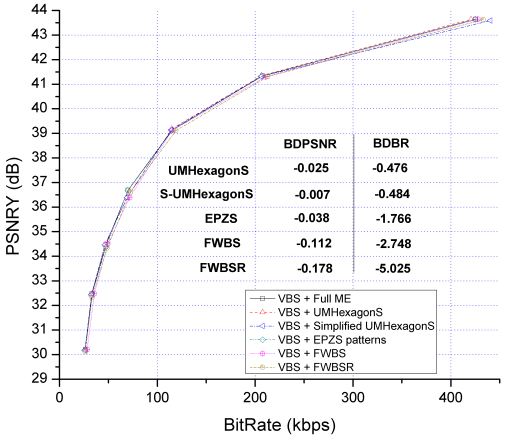
Block matching is used more frequently than any other ME technique in motion-compensated coding. It works based on partitioning a frame into non-overlapped, equally spaced, fixed size, small rectangular blocks and assuming that all pixels in a block experience the same translational motion. Consequently, block matching is much simpler and involves less side information compared with ME for arbitrarily shaped blocks. In this Chapter, various issues related to block matching are discussed such as selection of block sizes, cost functions, and search algorithms. The SAD is commonly used as a cost function, it also provides a bottleneck for fast ME algorithm. We propose the FWBS algorithm based on



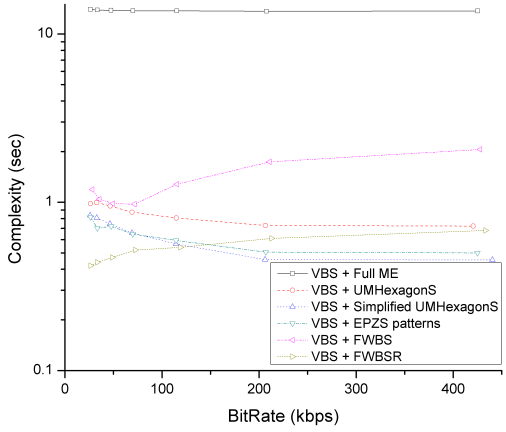
(a) R-D performance comparison with the other fast MEs implemented in the JM; VBS+FWBS shows almost the same performance as the other algorithms. VBS+FWBSR has considerable PSNR drop of low bit rate; Red box represents PSNR drop for VBS+FWBSR, yellow box represents PSNR drop for VBS+FWBS compared to JM



(b) C-R performance comparison with the other fast MEs ; VBS+FWBS shows slightly higher complexity than the other algorithm. VBS+FWBSR shows the minimum complexity among algorithms



(c) R-D performance comparison with the other fast MEs implemented in the JM; VBS+FWBS and VBS+FWBSR show almost the same performance as the other algorithms in static sequences



(d) C-R performance comparison with the other fast MEs

Figure 6.6: R-D and C-R performance comparison for (a)&(b) Foreman@352 $\times$ 288, (c)&(d) Mother and Daughter@352 $\times$ 288

the fact that the SAD of pixels is bounded as a infinity-norm and L-2-norm of transformed coefficients. Moreover, the relationship between a block and its sub-blocksŠ coefficients is presented, which gives an additional benefit to obtain transformed coefficients of a large block without an inverse transform. Moreover, this approach improves a computational efficiency by a factor 6.7 times to obtain the lower 16 coefficients of a 16 $\times$ 16 block compared to the fast algorithm. The FWBS outperforms over other commonly used fast algorithms in both computational complexity and MSE. The computational efficiency of the FWBS mainly comes from reducing the number of SAD operations not the

number of search points, which gives a good results on challenging data such as camera rotation. In the next chapter, a skip MB detection algorithm based on the FWBS is presented to further reduce computational complexity.

—If a man takes no thought about what is distant, he will find sorrow near at hand.

Confucius

7 Skip Macro-Block Detection

7.1 Introduction

The encoder normally skips a significant proportion of MBs, especially for sequences with low activity. Moreover, more skipped blocks appear as QP is increased as shown in Figure 5.6. If a MB is classified as a skipped MB, it means that no further encoding is carried out and a significant complexity reduction is possible since it is not necessary to perform ME, DCT, IDCT, and entropy coding. Skip MB detection algorithms have been researched for low complexity encoding in H.263 [23, 101], and H.264/AVC [55, 120, 128]. Since H.264/AVC in particular requires high computational complexity, skip MB detection is an attractive technique for reducing the computational complexity. Therefore we focus on a skip MB detection algorithm in the case of a H.264/AVC encoder in this chapter. In a H.264/AVC encoder, such as the official reference software (JM reference software), skip MB detection is carried out after encoding all the possible modes, this has no advantage in computational complexity. The proposed skip MB detection algorithm works with the VBS partitioning and the fast ME algorithm explained in Chapter 5 and Chapter 6. Our proposed skip mode detection uses the model of motion compensated pixel value presented in [18]. However, we cannot directly use it because our approach is performed in the SWHT domain and thus this approach is modified. The basic idea of the algorithm can be summarized as follows with details explained in this chapter:

- We first derive the relationship between the ICT and the SWHT by introducing simple orthogonal transform (named S-transform).

- Shifting the average value of motion compensated pixels by 3σ in the pixel domain. This only affects the DC coefficients of the ICT.
- If pixel values are positive real, then the other AC samples of the SWHT is bounded to $\pm\frac{1}{2}$ of the zero sequency term.
- In the ICT of 3σ shifted pixel values, the DC value could be the maximum value of the block. If the quantised DC is a zero, this block is considered as a Zero Quantized DCT coefficients detection (ZQDCT).

7.2 Related Work

The skip MB detection algorithm consists of two main components in general, ZQDCT and Zero Motion Detection (ZMD). Pixels in the blocks of a motion compensation frame are very close to zero. As expected, these blocks have great probability to be all-zero DCT coefficients blocks after DCT and quantisation. ZQDCT mainly focuses on detecting all-zero DCT coefficients blocks before DCT and quantisation, which makes the encoder more efficient. The ZMD is carried out to find blocks with zero motion vector, $mv(d_x, d_y) = (0, 0)$, which means that the best matched MB of the current frame is the same location of the reference frame [127]. Although ZMD methods are developed for previous coding schemes, such as H.263, they cannot be directly applied to H.264/AVC. This is because compared to H.263, where only two block sizes (16×16 and 8×8) are used, seven block sizes varying from 16×16 to 4×4 are used in H.264/AVC. In order to adapt the ZMD algorithm to H.264/AVC, the most common approaches focus on the pre-defined threshold empirically obtained based on R-D optimization [20, 128]. This introduces computational complexity as well as the possibility of quality degradation by selecting an inappropriate threshold. Moreover, H.264/AVC adopts motion vector prediction using neighboring blocks based on the fact that the motion vector has a close correlation with neighboring blocks in the spatial domain. Thus research on skip MB has been more focused on finding ZQDCT blocks with accurate and low complexity algorithms.

In [124], an early detection method for ZQDCT was proposed by defining a sufficient condition for quantising all DCT coefficients to zero. Each block is checked for this condition, and DCT and quantisation are skipped if it holds. In [98, 118], the authors theoretically derived a precise condition and improved

[124]'s algorithms. Another ZQDCT algorithm was presented based on the theoretical analyzes of the ICT and quantisation in H.264/AVC [76], where they proposed the relationship between SAD and threshold value. In [117, 119], another sufficient condition related to SAD was presented for predicting ZQDCT before DCT and quantisation to reduce redundant DCT and quantisation computations. A spatiotemporal characteristic of the R-D cost function and SAD was proposed in [39], where they used partially computed SAD to reduce computational complexity. A model-based skip MB detection algorithm was presented in [18], where the pixel value of motion compensated blocks was modeled as a generalized Gaussian distribution according to QP and selected threshold values.

H.264/AVC optionally supports SATD. It works by taking the SAD of the WHT on 4×4 blocks. The SATD is much slower than the SAD, which is a critical drawback. The benefit of the SATD is that it more accurately predicts quality from both the standpoint of objective and subjective metrics. In this case, the ZQDCT provided in [76, 117, 119] can not be applied. In [120], they first presented a ZQDCT algorithm by utilizing the SATD, where a threshold based criterion for ZQDCT prediction is used. Their results show a comparable performance for both R-D and complexity savings. However, threshold values should be obtained by calculating by floating point matrix operations, where the performance could be affected by the rounding error of threshold values. Moreover, those thresholds are compared with all coefficients in a block, which introduces computational complexity.

7.3 Relationship Between the Integer DCT (ICT) and the SWHT

7.3.1 Integer DCT

H.264/AVC uses 4×4 or 8×8 ICT instead of the 8×8 DCT used in MPEG-2. The ICT is another version of the DCT with lower complexity and little performance degradation by introducing integer operations [69]. It only involves additions and shift operations and no mismatch exists between the forward and inverse transform. We consider the 4×4 ICT in this section because the 8×8 ICT is only used for the Fidelity Range Extensions [104].

A 4×4 DCT is given by

$$Y = DXD^T = \begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix} X \begin{bmatrix} a & b & a & c \\ a & c & -a & -b \\ a & -c & -a & b \\ a & -b & a & -c \end{bmatrix} \tag{7.1}$$

where :

$$a = \frac{1}{2}, \quad b = \sqrt{\frac{1}{2}}\cos\left(\frac{\pi}{8}\right), \quad c = \sqrt{\frac{1}{2}}\cos\left(\frac{3\pi}{8}\right)$$

This matrix multiplication can be factorized to the following equivalent form:

$$\begin{aligned} Y &= (CXC^T) \otimes E \\ &= \left(\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & d & -d & -1 \\ 1 & -1 & -1 & 1 \\ d & -1 & 1 & -d \end{bmatrix} X \begin{bmatrix} 1 & 1 & 1 & d \\ 1 & d & -1 & -1 \\ 1 & -d & -1 & 1 \\ 1 & -1 & 1 & -d \end{bmatrix} \right) \otimes \begin{bmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{bmatrix} \end{aligned} \tag{7.2}$$

where E is a matrix of scaling factors and the symbol $\otimes$ represents that each element of CXC^T is multiplied by the scaling factor in the same position in matrix E, and d is $\frac{c}{b}$. To simplify the implementation of the transform, d is approximated by 0.5. In order to ensure that the transform remains orthogonal, b and a should be chosen so that:

$$a = \frac{1}{2}, \quad b = \sqrt{\frac{2}{5}}, \quad d = \frac{1}{2} \tag{7.3}$$

Moreover, the post-scaling matrix E is scaled down in order to avoid multiplications in the transform CXC^T. The final Integer DCT of a 4×4 block becomes:

$$\begin{aligned} Y &= (C'XC'^T) \otimes E' \\ &= \left(\begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} X \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 2 \\ 1 & -2 & 1 & -1 \end{bmatrix} \right) \otimes \begin{bmatrix} a^2 & \frac{ab}{2} & a^2 & \frac{ab}{2} \\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4} \\ a^2 & \frac{ab}{2} & a^2 & \frac{ab}{2} \\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4} \end{bmatrix} \end{aligned} \tag{7.4}$$

This transform is an approximation to the 4×4 DCT. The result of the transform is not identical to the 4×4 DCT because the DCT performs floating point not integer operations.

7.3.2 Quantisation in H.264/AVC

The mechanism of the forward and inverse quantizers in H.264/AVC is implemented by the requirements to avoid division and floating point operations. It incorporates the pre-scaling matrices, E', described in Equation (7.4). If the quantised coefficients, Z_{ij}, and the transformed coefficients, Y_{ij}, and quantizer step size, Q_{step} are defined, the relationship between Z_{ij} and Y_{ij} is noted as

$$Z_{ij} = round\left(\frac{Y_{ij}}{Q_{step}}\right). \tag{7.5}$$

A total of 52 values of Q_{step} are supported by the standard, indexed by a QP. The QP was designed to be doubled when the QP increases every 6. The wide range of quantizer step sizes makes it possible for an encoder to control the trade-off between bit rate and quality accurately and flexibly. The post-scaling factor (PF) is incorporated into the forward quantiser, thus Equation (7.5) can be rewritten as

$$Z_{ij} = round\left(C_{ij}\frac{PF}{Q_{step}}\right) \tag{7.6}$$

where C_{ij} represents coefficients of the core transform at position (i, j) denoted as $C'X'C'^{T}$ in Equation (7.4). In order to simplify, the factor $\frac{PF}{Q_{step}}$, is implemented as a multiplication by a factor MF and shift operations, avoiding any division operation;

$$Z_{ij} = round\left(C_{ij}\frac{MF}{2^{qbits}}\right) \tag{7.7}$$

where

$$\frac{MF}{2^{qbits}} = \frac{PF}{Q_{step}}, \quad qbits = 15 + floor(QP/6)$$

In integer operations, Equation (7.7) can be implemented as

$$Z_{ij} = sign(C_{ij}) \times (|C_{ij}| \times MF + f) >> qbits \tag{7.8}$$

where » indicates a binary shift operation. f is defined as $2^{qbits/3}$ for Intra blocks or $2^{qbits/6}$ for Inter blocks in the JM reference software, where the quantisation related variables are defined as a LookUp Table (LUT) as follows:

$$Z_{ij} = sign(C_{ij}) \times |C_{ij}| \times \frac{2^{qbits} - qp\_const}{quant\_coef[qp\_rem][i][j]} \tag{7.9}$$

where $qp\_rem = QP\%6$, $qbits = QP/6 + 15$, $qp\_const = (1 << qbits)/6$, and $quant\_coef$ is the scaling matrix.

7.3.3 Relationship between ICT and SWHT

Let the matrices of the ICT and the SWHT be C' and W respectively, that is

$$C' = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}, \quad W = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}. \tag{7.10}$$

From their relationship, we obtain the S matrix, which also satisfies the orthogonal condition like the ICT.

$$C' = S \times W, \quad S = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 6 & 0 & 2 \\ 0 & 0 & 4 & 0 \\ 0 & -2 & 0 & 6 \end{bmatrix} \tag{7.11}$$

Equation (7.4) can be rewritten as

$$\begin{aligned} Y = C'XC'^T \otimes E' &= SWX(SW)^T \otimes E' \\ &= SWXW^TS^T \otimes E' \\ &= \frac{1}{4}S\mathbf{X}S^T \otimes E' \\ &= \frac{1}{16}\mathcal{X} \otimes E' \\ &= \mathcal{X} \otimes E'' \end{aligned} \tag{7.12}$$

where $\mathbf{X}$ is a SWHT of a 4×4 block, $\mathcal{X}$ represents the S-transform of the SWHT coefficients, and E' is scaled down to E'' by $\frac{1}{16}$ due to the normalized factor of the two transforms (SWHT and S-transform).

$$E'' = \frac{1}{16}E' \tag{7.13}$$

7.4 Zero Quantised DCT Coefficients Detection

Suppose the motion compensated residual pixel values x_i at the input of the ICT are approximated by a Gaussian distribution with zero mean and variance σ^2 [18],

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}, \quad -\infty < x < +\infty. \tag{7.14}$$

The expectation value of $|x|$ can be calculated as

$$E[|x|] = \int_{-\infty}^{+\infty} |x| \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} dx = \sqrt{\frac{2}{\pi}}\sigma \tag{7.15}$$

Let the sum of the absolute value of a 4×4 block be β, $\beta = \frac{1}{16}\sum_{i=1}^{16} |x_i|$, we obtain σ from the relationship between β and $E[|x|]$, that is

$$\sigma = \sqrt{\frac{\pi}{2}}\beta \tag{7.16}$$

Since SAD values are not used in our proposed method, we apply [119]'s method to obtain σ in the WHT domain, where $\alpha = \frac{1}{16}\sum_{i=1}^{16} |w_i|$, w_i is a i^{th} SWHT coefficient in a 4×4 block.

$$\sigma = 2 \times \sqrt{\frac{\pi}{2}}\alpha = \sqrt{2\pi}\alpha \tag{7.17}$$

Let the Gaussian distribution be shifted by 3σ, then the probability for pixel values to be positive is

$$P(x > 0) = \int_{0}^{+\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-3\sigma)^2}{2\sigma^2}} dx = 99.9\% \tag{7.18}$$

Note that adding 3σ to a compensated residual data makes all data positive with 99.9% probability, which only affects the DC or zero sequency term in the ICT or the SWHT. Figure 7.1 illustrates this graphically. From Equation (A.6), the ICT of the 4×4 block can be obtained from the SWHT coefficients ($\mathbf{W} = \{w_0, \ldots, w_{15}\}$) as shown in Figure 7.2. In order to show that the DC value of Y is the maximum value, the DC value $16 \times w_0$ is compared with the other AC coefficients considering scaling factor($E''(i,j)$), where (i,j) is a position corresponding to a row (i) and column (j) of E''. Firstly, the DC value of Y is

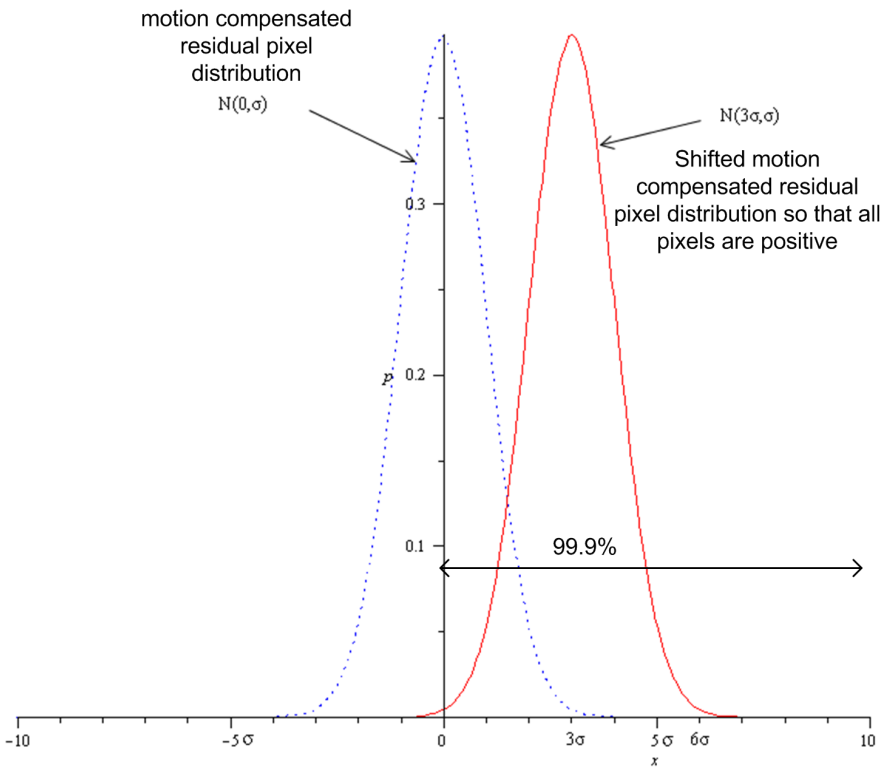


Figure 7.1: 3σ shifted motion compensated residue data; probability of pixel value to be positive is 0.999

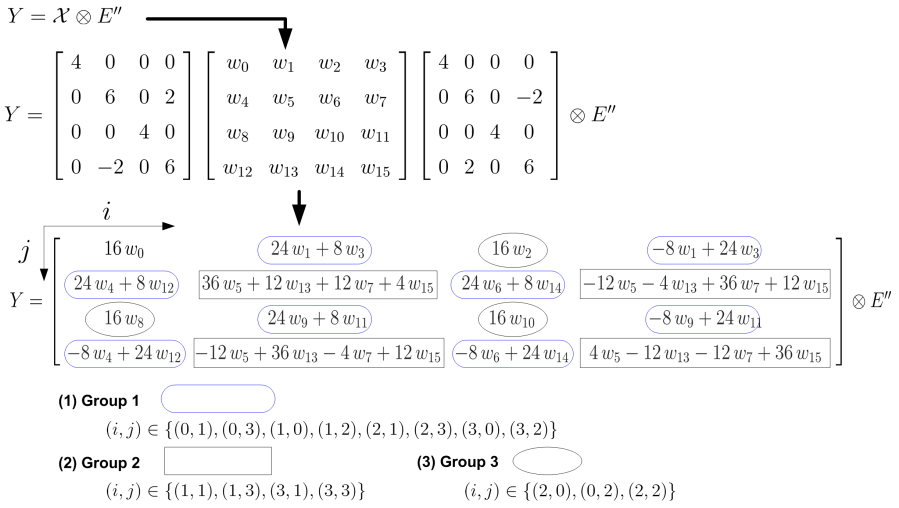


Figure 7.2: Illustration of comparison between DC and the other AC coefficients

compared with AC coefficients located in

$$(i,j) \in \{(0,1),(0,3),(1,0),(1,2),(2,1),(2,3),(3,0),(3,2)\}$$

denoted as a Group 1 in Figure 7.2. The scaling factor, $\frac{E''(i,j)}{E''(0,0)}$, at Group 1 can be obtained as $\frac{b}{2a} \approx 0.63$ by observing Equation (7.4). If $x(i,j)$ is a positive real value, then the maximum possible value for the zero sequency term is N^2A where A is the maximum value of $x(i,j)$. All Hadamard domain samples other than the zero sequency range between $\pm\frac{N^2A}{2}$. The magnitude of the zero sequency term is a bound for the magnitude of all other Hadamard domain samples as mentioned in [88] so that

$$w_k \leq \left|\frac{w_0}{2}\right|, \quad k \neq 0. \tag{7.19}$$

The comparison between the DC and the Group 1 coefficients can be denoted as

$$\begin{aligned}(24w_a \pm 8w_b) \times \frac{E''(i,j)}{E''(0,0)} &\leq (24|w_a| + 8|w_b|) \times \frac{E''(i,j)}{E''(0,0)} \\ &\leq \left(24\left|\frac{1}{2}w_0\right| + 8\left|\frac{1}{2}w_0\right|\right) \times \frac{E''(i,j)}{E''(0,0)} \\ &= 10.08w_0 < 16w_0\end{aligned} \tag{7.20}$$

where w_a and w_b represent the SWHT coefficients of Group 1. Therefore, the DC value is always larger than any other coefficients involved in Group 1. Secondly, we consider the other positions, $(i,j) \in \{(1,1),(1,3),(3,1),(3,3)\}$, denoted as Group 2 in Figure 7.2. The scaling factor, $\frac{E''(i,j)}{E''(0,0)}$, is $\frac{b^2}{4a^2} = 0.4$. Therefore, the comparison between the DC value and Group 2 AC coefficients can be given as

$$\begin{aligned}(36w_a \pm 12w_b \pm 12w_c \pm 4w_d) \times \frac{E''(i,j)}{E''(0,0)} & \\ &\leq (36|w_a| + 12|w_b| + 12|w_c| + 4|w_d|) \times \frac{E''(i,j)}{E''(0,0)} \\ &\leq 32w_0 \times \frac{E''(i,j)}{E''(0,0)} = 12.8w_0 < 16w_0\end{aligned} \tag{7.21}$$

Thus, the DC value is always larger than any other AC coefficients of Group 2. Finally, the DC value is also larger than any other AC coefficients of Group 3 from Equation (7.19). Therefore, the DC of Y is always greater than the other coefficients. When the DC of the ICT after quantisation is zero, this block is considered as a zero coefficients' block due to the fact that the DC is the

maximum coefficient of the block by Equation (7.20) and Equation (7.21). The overall procedure to obtain the ICT of 3σ shifted motion compensated pixels is summarized as

$$\begin{aligned}
&(x_0,\ldots,x_{15})+(3\sigma,\ldots,3\sigma)\\
&\Downarrow\Downarrow \text{SWHT}\\
&(w_0,\ldots,w_{15})+(4\times 3\sigma,0,\ldots,0)\\
&\Downarrow \text{S}-\text{transform}\\
&(16(w_0+4\times 3\sigma),24w_1+8w_3,\ldots,4w_5-12w_{13}-12w_7+36w_{15}) \quad = C_{ij}
\end{aligned} \tag{7.22}$$

where C_{ij} represents the ICT coefficient of a 4×4 block at position (i,j). When the DC coefficient of the ICT after applying quantisation is a zero as shown in Equation (7.23), this block is considered as a ZQDCT.

$$\left(16w_0+16\times 4\times 3\sigma\right)Q'_{00}=\left(w_0+12\sqrt{2\pi}\sum_{k=0}^{15}|w_k|\right)Q_{00}=C_{00}\times Q'_{00}=0 \tag{7.23}$$

where Q'_{00} is $\frac{Q_{00}}{16}$ by observing that scaling factor $E''=\frac{E'}{16}$ as shown in Equation (7.13), and Q_{00} is a quantisation step of DC coefficient defined in JM as:

$$Q_{00}=\left(\frac{2^{qbits}-qp\_const}{quant\_coef[qp\_rem][0][0]}\right).$$

7.5 Skip Macro-block Detection

7.5.1 Detection Algorithm

In H.264/AVC, the requirements to be a skipped block are as follows.

1. MB : The block must be a 16×16 block.
2. ZMD : Motion vector are predictive motion vectors, pmv, using the median value of neighboring blocks' motion vectors.
3. ZQDCT : All DCT coefficients after quantisation are zeros.

The proposed skip MB detection algorithm works on a 16×16 not a 4×4 block. It is possible to detect ZQDCT by dividing a MB to 4×4 blocks. However,

it requires recalculating the 4×4 blocks' SWHT, which definitely introduces computational complexity and more intermediate memory. Thus, we need to modify Equation (7.23) to fit for a 16×16 block. A 16×16 block is considered as a skip MB if all 4×4 blocks in the 16×16 block are ZQDCTs.

Let the SWHT coefficients of a 16×16 block be

$$\mathbf{W^{16}} = \{W^{16}(0), W^{16}(1), \ldots, W^{16}(16 \times 16 - 1)\}$$

and its i^{th} sub partitioned 4×4 sub-block' coefficients be

$$\mathbf{W_i^4} = \{W_i^4(0), W_i^4(1), \ldots, W_i^4(15)\}$$

from the raster scan order. We assume that σ of a 4×4 block is the same as that of a 16×16 block given as:

$$\sigma = \sigma_{4\times4} = \sigma_{16\times16} \tag{7.24}$$

From Equation (7.23), ZQDCT is decided when all sub partitioned 4×4 blocks' coefficients are zeros after quantisation. Thus, metric (M) is as follows;

$$M = \left(\sum_{k=0}^{15} W_k^4(0) + 16 \times 3\sigma\right) Q_{00}. \tag{7.25}$$

From Equation (7.17), the σ of a 16×16 block can be obtained by using 16 lower coefficients instead of using all coefficients of a 16×16 block because a few coefficients have most of the energy in a block, which is given as:

$$\begin{aligned} \sigma &= \sqrt{2\pi}\frac{\sum_{k=0}^{k=255}}{16 \times 16}|W^{16}(k)| \\ &= \sqrt{2\pi}\frac{\sum_{k=0}^{k=15}}{16 \times 16}|W^{16}(k)|. \end{aligned} \tag{7.26}$$

From Lemma 6.2 in Chapter 6, the zero sequency term of a 16×16 WHT can be obtained using sub blocks' zero sequency terms as follow;

$$W^{16}(0) = \frac{\sum_{k=0}^{15} W_k^4(0)}{4} \tag{7.27}$$

From Equation (7.26) and Equation (7.27), Equation (7.25) can be rewritten as follows;

$$\begin{aligned} M &= \left(\sum_{k=0}^{15} W_k^4(0) + 16 \times 3\sigma \right) Q_{00} \\ &= \left(4W^{16}(0) + \frac{16 \times 3\sqrt{2\pi}}{16 \times 16} \sum_{k=0}^{k=15} |W^{16}(k)| \right) Q_{00} \\ &= \left(4W^{16}(0) + 0.47 \times \sum_{k=0}^{k=15} |W^{16}(k)| \right) Q_{00} \\ &\cong \left(4W^{16}(0) + \left(\sum_{k=0}^{k=15} |W^{16}(k)| \right) >> 1 \right) Q_{00} \end{aligned} \tag{7.28}$$

When Equation (7.28) is zero, this block is considered as a skip MB, which means that all coefficients of a 16×16 block are zeros. Therefore, after checking the position at pmv in the middle of ME, the zero sequency term $W^{16}(0)$, and α are calculated. Finally, when M mentioned in Equation (7.28) is zero, this block is classified as a skip MB.

7.5.2 Results

In order to evaluate the proposed approach, the JM 11.0 is used for experiments. Tests are performed with the following encoder configuration; (1) GOP has IPPP structures without B-frames, (2) The inter QPs are selected seven values, 20,24,28,32,36,40, and 44, and the intra QP is the same as inter QP. Four benchmark video sequences are used "Foreman", "Mother and Daughter", "Pedestrian", and "Rush hour" with QCIF to 720p-HD format.

The skip MB detection algorithm needs to begin with full ME, which is most computationally complex function in the encoder side. skip MB detection is performed only for 16×16 blocks. The skip MB is performed starting from the pmv position. When Equation (7.28) is zero at this position, no further processing including ME is needed.

For evaluation, the Precision rate (PR) and the false acceptance rate (FAR) are introduced as follows:

$$PR = \frac{N_s}{N_s'} \times 100\%, \qquad FAR = \frac{(N_s \cap N_{ns}')}{N_s} \times 100\% \tag{7.29}$$

N'_s and N'_{ns} are the number of skip MBs and non skip MBs detected by the reference software respectively. N_s is the number of skip MBs obtained by the proposed method. It is desirable to have large PR and small FAR values for an efficient skip MB detection algorithm. In addition, the Percentage for Skip Macro-Block (PSM) is defined as

$$PSM = \frac{N'_s}{N'_m} \times 100\% \tag{7.30}$$

where N'_m represents the total number of MB. The encoded video quality and bit rates are objectively evaluated in terms of the PSNR ($\Delta P(dB)$) and bit rates saving (ΔR) presented in the following form:

$$\Delta P = P_{JM} - P_{proposed}, \qquad \Delta R = \frac{R_{proposed} - R_{JM}}{R_{JM}} \times 100\% \tag{7.31}$$

where $P_{proposed}$ and P_{JM} are the PSNR of the proposed approach and the JM reference encoder respectively; $R_{proposed}$ and R_{JM} are the encoded bit rates of the proposed approach and the JM encoder respectively. Finally, the computational complexity of overall encoding time improvement ΔT is observed via the following criteria:

$$\Delta T = \frac{T_{JM} - T_{proposed}}{T_{JM}} \times 100\% \tag{7.32}$$

The PR and FAR results are given along with the PSM in Table 7.1. From the results, the following conclusions can be drawn. Firstly, as skip MB occupy a great portion of the whole sequence as QP increases, more blocks can be determined as skip MBs as shown by PSM. Secondly, with an increase in QP, the proposed approach is able to predict skip MBs more efficiently-this can be clearly seen by observing PR. However, PR does not increase according to QP in high motion sequences such as Foreman because of the relatively low ratio of skip MBs for the whole sequence (note that PR does not increase in accordance with QP). Thirdly, as for the FAR result, the FAR becomes a little bit worse with an increase of QP. However, since the value of $(N_s \cap N'_{ns})$ is relatively small as compared with N_s, the improvement in terms of the PR becomes more dominant. The FAR of the proposed approach results in insignificant video quality degradation as seen by the noted ΔP. In a nutshell, the ΔT is an index of how much the complexity of the encoder is reduced. On the contrary, the ΔP represents the degradation of encoded video quality. Finally, the video quality is objectively evaluated in terms of the PSNR and bit-rate. From the

Table 7.1: Performance comparison of proposed approach to JM

Sequences		QP	PSM (%)	PR (%)	FAR (%)	Δ P (dB)	Δ R (%)	Δ T (%)
QCIF 176x144	Foreman	20	3.6	33.3	0.10	0	0	1.2
		24	8.7	12.5	0.17	0.06	0	2.2
		28	17.6	11.7	0.43	0.08	+0.25	2.32
		32	32.3	12.5	0.58	0.04	+0.24	4.97
		36	57.2	15.2	1.21	0.09	+1.62	5.24
		40	77.3	11.5	2.31	0.09	+1.13	10.1
		44	93.5	19.4	2.89	0.27	+7.47	19.5
	Mother & daughter	20	43.8	14.0	0.23	0	0	6.8
		24	50.5	14.2	0.18	0.01	+0.21	9.2
		28	63.9	15.9	0.26	0.19	+0.46	12.4
		32	81.2	13.6	0.31	0.01	+0.67	12.5
		36	93.2	14.9	0.31	0.02	0	16.9
		40	99.2	28.3	0.34	0.16	+1.02	27.8
		44	100.0	63.3	0.30	0.15	+6.44	61.0
CIF 352x288	Foreman	20	4.0	40.2	0.24	0	-0.10	3.7
		24	10.2	30.0	0.36	0.02	+0.31	5.4
		28	24.4	20.8	0.52	0.08	+0.83	7.4
		32	47.3	27.7	1.02	0.19	+0.94	17.7
		36	74.3	24.3	1.21	0.19	0	23.3
		40	89.2	31.5	1.41	0.27	+0.08	33.7
		44	96.3	40.6	1.55	0.29	+0.15	40.2
	Mother and daughter	20	39.2	38.5	0.18	0.12	+0.10	22.8
		24	54.3	38.8	0.22	0.05	-0.07	28.8
		28	67.5	34.5	0.24	0.29	+0.53	36.9
		32	84.9	34.5	0.41	0.08	+1.56	37.4
		36	95.4	35.9	0.48	0.22	+2.19	47.9
		40	99.1	50.2	0.51	0.20	+1.91	55.4
		44	100.0	78.8	0.51	0.17	+2.54	62.8
SD 720x576	Pedestrian	20	22.6	27.2	0.26	0.05	+0.07	0.9
		24	35.7	20.1	0.54	0.03	+0.09	1.9
		28	47.0	19.2	0.88	0.09	+0.21	6.3
		32	61.8	19.7	0.91	0.20	+0.51	8.4
		36	76.9	25.2	1.53	0.27	0	21.9
		40	88.5	33.4	1.91	0.41	+1.92	28.6
		44	97.8	43.3	2.34	0.48	+4.11	45.6
	Rush Hour	20	26.7	11.5	0.11	0.03	0	6.1
		24	41.8	14.6	0.28	0.03	+0.15	7.7
		28	55.7	23.6	0.75	0.09	+0.27	15.3
		32	71.1	29.6	1.53	0.19	+0.39	27.8
		36	85.5	36.5	1.72	0.24	+0.43	32.5
		40	94.7	47.9	2.66	0.43	+3.19	47.4
		44	98.7	62.2	2.25	0.52	+4.54	64.2
HD 1280x720	Pedestrian	20	22.9	22.7	0.36	0.03	+0.08	7.8
		24	38.0	18.4	0.54	0.05	+0.10	9.5
		28	52.9	21.2	0.91	0.15	+0.22	10.2
		32	67.9	22.4	1.07	0.21	+0.58	13.7
		36	82.9	24.7	1.53	0.33	+1.30	26.9
		40	93.1	33.7	2.31	0.45	+2.81	37.8
		44	98.2	46.9	2.61	0.59	+2.74	50.6
	Rush Hour	20	29.8	17.2	0.22	0.04	+0.06	7.3
		24	51.2	15.7	0.31	0.04	+0.22	12.1
		28	66.3	24.5	0.75	0.03	+0.21	22.4
		32	81.8	32.1	1.48	0.21	+0.77	32.6
		36	92.2	40.2	2.54	0.38	+1.61	44.3
		40	97.8	53.6	3.28	0.48	+3.98	57.9
		44	99.8	67.3	3.25	0.58	+7.59	70.1

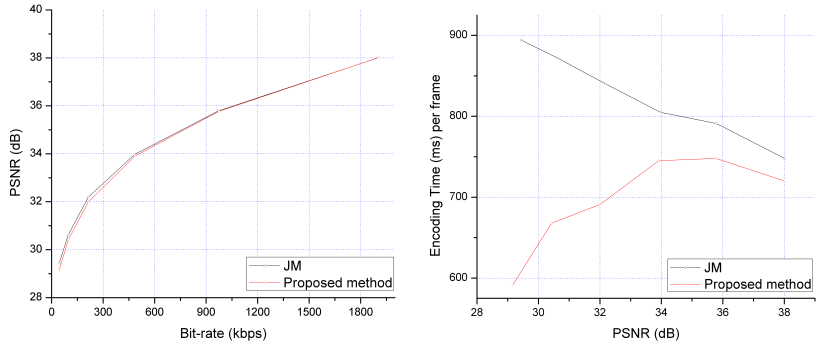
results, the maximum PSNR loss is 0.59dB at 720p-HD Pedestrian sequence at QP 44. Therefore, the PSNR drop of the proposed approach is negligible for all sequences. From a complexity perspective, the proposed approach can greatly reduce encoding time in accordance with QP since skip MBs are detected before performing ME, DCT and IDCT. When the block's DC of WHT at the initial position ($mv = (pmv_x, pmv_y)$) after quantisation is zero, no more ME is needed, which is the main contributive factor for reducing encoding time. As QP increases, more skip MBs are detected, further reducing computational complexity. The proposed approach can reduce the overall encoding time by 1.2%-70.2% at various conditions as shown in Table 7.1.

The commonly used skip mode detection in H.264/AVC is performed after deciding VBS partitioning based on a mode competition. It has no computational advantage when an image is encoded with a high value of QP as shown in Figure 7.3(b)(d). When the proposed skip MB detection algorithm is applied to JM for "Foreman" and "Mother and Daughter" sequence as shown in Figure 7.3, C-D shows a good performance especially in high QP (see Figure 7.3(b)(d)) whilst the degradation of R-D is negligible (see Figure 7.3(a)(c)).

7.6 Discussion

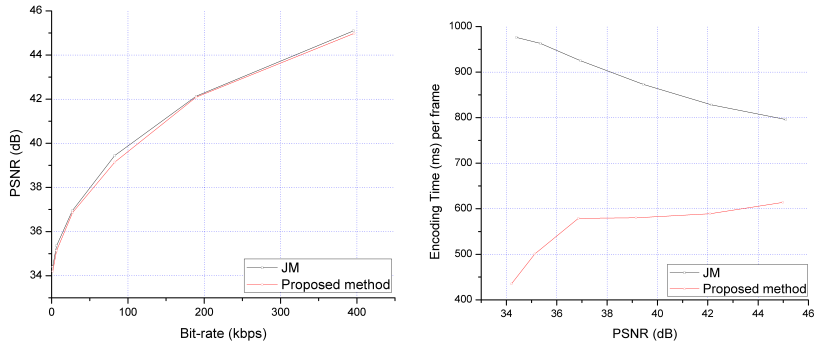
In video compression, ME, DCT and IDCT need large amounts of computation, so it is desirable to reduce the time required for conducting ME, DCT and IDCT for most video encoders, especially for power limited portable video codec devices. Pixels in the blocks of motion compensation frame are very close to zero. As expected, these blocks have great probability to be all-zero DCT coefficients after quantisation, which are classified as skip MBs with two more conditions in H.264/AVC; zero motion vector and a 16×16 block. Skip MBs do not require any encoding procedure, so huge computational complexity saving is possible if they are detected accurately.

In this chapter, a skip MB detection algorithm based on the SWHT is presented. A simple transform (S-transform) is also proposed by observing the relationship between ICT and SWHT. And 3σ shifting of the mean value in motion compensated frame, which is modeled as a Gaussian Distribution, makes all



(a) R-D performance comparison between JM and the proposed method

(b) Encoding time vs. PSNR; much computational savings are achieved in low PSNR (high QP)



(c) R-D performance

(d) Encoding time vs. PSNR

Figure 7.3: R-D and C-D performance for "Foreman" and "Mother and Daughter" with CIF

pixel values of the compensated frame positive. The necessary and sufficient condition to be a skip MB is derived by observing the relationship between ICT and SWHT on the shifted values. As shown in Table 7.1, the proposed approach can greatly reduce encoding time and achieves almost the same R-D performance as the JM reference encoder. In the following chapter, a complexity adapted video encoder framework is presented based on controlling the number of skip MBs.

—Liberty without learning is always in peril and learning without liberty is always in vain.

John F. Kennedy

8 A Framework for Complexity Adaptation in a Video Encoder

8.1 Introduction

Power consumption is an important issue in a power limited platform. In the case of multimedia applications, the battery life of such devices has been shown to be directly connected to the computational complexity of the associated data processing function. However, computational complexity and video quality are not comfortable bed-fellows. Therefore, for real-time video coding applications, it is important to be able to control the complexity of the encoder without significantly affecting R-D performance. Moreover, a video coding algorithm that gives excellent visual quality for a given bit rate might be impractical if it requires too much computational complexity. In [99, 100], the characteristics of R-D are shown to be almost equal to that of C-D for infinite observed stationary-ergodic sources. Therefore, optimization in terms of computational complexity could be obtained via RDO techniques such as Lagrangian multiplier method [58, 121] and Dynamic Programming (DP) [11]. Recent trends in video codec design require a more flexible approach to trade-offs between complexity and quality especially for software or power limited video codecs. Of course complexity has a close connection with bit rate which means that optimization should be performed based on three terms: complexity, bit rate, and quality. However, it is difficult to find the theoretical optimum point since the relationship between the three variables is affected by various factors such as coding parameters, and low complexity algorithms. Therefore, research mainly focuses on

complexity and distortion whilst keeping reasonable R-D performance. Widely used complexity control algorithms in H.264/AVC can be classified as follows;

1. Frame skipping: Skipping frames is an effective way of reducing processor utilization. When the frame rate is low, there will be a large difference between successive frames, which leads to more bits on the residual frame and to the degradation of video quality. Frame skipping is mainly used in a static scene such as video telephony or surveillance applications.

2. Motion search range control: ME is the most time consuming function in a video coder. If the search range increases, the residual data becomes smaller. For some sequences, increasing the search range will not lead to improved performance. The computational complexity savings of fast algorithms are achieved by reducing the search range.

3. Multiple reference frame control: H.264/AVC uses multiple reference frames to obtain good coding performance. However, complexity increases in proportion to how many reference frames are used.

4. Skip MBs and zero motion detection: In a DCT-based codec at medium or low bit rates, many blocks contain no AC or DC coefficients after quantisation. If zero quantised coefficients are detected prior to ME, DCT and IDCT, huge computational complexity saving can be achieved.

In Chapter 5 and Chapter 6, a FWBS algorithm based on VBS is presented. In Chapter 7, a skip MB detection algorithm is also proposed. In this chapter, we present the framework for complexity adaptation in a video coder by combining the above techniques into the JM reference software. The FWBS based on VBS, skip MB, and zero motion detection algorithm are integrated into the JM reference software and tested. Finally, a C-D model is proposed for adapting the computational complexity of a video coder.

8.2 Related Work

A complexity adaptation algorithm is typically based on a complexity reduction algorithm, which achieves varying degrees of complexity savings depending on

the statistics of the source video. In computation or power constrained applications, it is important to able to control and manage the computational complexity of key components in the video encoder. Moreover, to maximize R-D performance, the most commonly used method is a high complexity RDO mode selection process, which includes encoding the MB in all possible modes and finding the minimum R-D cost function. It gives significant improvement in video coding performance. However, it requires intensive computational complexity, so it is difficult to apply for real-time applications and video applications on complexity constrained platform. A significant amount of research has focused on developing low complexity implementations of mode selection, ME and DCT, which account for most of complexity of an encoder. Little research has been carried out on issues of computational complexity management. Prior research can be classified in several ways;

1. Algorithms that achieve the required computational complexity by reducing the complexity of ME and DCT, which are the key burdens of an encoder. Tai *et al.* [106] presented a software-based computation-aware scheme that terminates the searching process if a pre-defined computation has been reached, where more computation is allocated to the MB with larger distortion in a step-by-step fashion. In [17], an extended version of [106] was proposed to reduce the memory required and to use the context information of neighboring blocks. In [4], a complexity scalable and control algorithm in H.264/AVC was proposed. The complexity is adapted jointly by parameters that determine the aggressiveness of an early stopping criterion. Moreover, Ates *et al.* [7] presented a joint R-D and complexity framework for ME using the spatiotemporal gradient of each MB. They showed that coding performance in terms of R-D and complexity could be reduced if prior knowledge of video characteristics such as gradients is available. This motivated our proposed VBS partitioning algorithm and its results show reasonable performance as explained in Chapter 5. The above approaches focused on reducing the number of SAD operations during ME. They therefore provide computational complexity control over full ME methods only. When a fast algorithm is applied, there is no scope for controlling computational complexity. Unfortunately whilst fast algorithms should be used for real-time or video applications for power limited platforms, these algorithms cannot be adapted properly.

2. Algorithms that maximize R-D performance for given complexity called C-R-D or Power-Rate-Distortion (P-R-D). He *et al.* presented a P-R-D model under a energy constraint especially targeting wireless video communication. They determined R-D behavior at a given complexity control parameter using the linear R-D model suggested in [32]. Computational complexity control is achieved by finding parameters using frame skipping and controlling the SAD operations of ME. In [111], joint C-R-D analysis of H.264/AVC was presented, where they suggested an algorithm called GBFOS that chose the right set of encoder parameters. Their extended research was also described in [112]. However, their algorithms fundamentally require iterative operations to find suitable parameters. This means that pre-decided parameters obtained by off-line simulation are needed for real-time operation.

3. Algorithms that control complexity using a Lagrangian cost function. A Lagrangian cost based skip prediction algorithm was presented in [41], where complexity management was performed by adjusting the cost of skip prediction. The results show good complexity adaptation properties. In [34], a joint C-R-D for ME was proposed by observing two Lagrange parameters used to cut off complexity inefficient motion search. They used fitting curves to represent the complexity-parameter, which controls the Lagrange multiplier allowing complexity to be controlled.

The most common approaches to control complexity focus on ME, mode decision, and DCT. However, common approaches require pre-calculated parameters or iterative operations to find the best parameter set. This is a key bottleneck for real-time operation on power limited platforms, especially without prior-knowledge of sequence. Therefore, a complexity adaptation algorithm based on a complexity model is presented in this chapter. The proposed algorithm has two distinctive characteristics; (1) A frame level complexity control algorithm based on a complexity model. (2) No need for feedback information or pre-defined parameters, which give a benefit in case of managing unknown parameters of sequences. In [100], it was shown that R-D is almost surely equal to C-D for stationary-ergodic sources. Moreover, He *et al.* [31] presented R-D analysis by introducing the ρ domain. ρ is defined as the percentage of zero DCT coefficients in a frame. Motivated from [31, 100], the proposed algorithm used a similar approach to build up a model. However, we would like to note

that the proposed complexity adaptation algorithm uses totally different parameters (bitrate (R)-zero coefficient (ρ) vs. complexity (C)-skip block ratio (ρ) and quantisation parameters (QP) vs. threshold value for skip blocks (τ)) compared to the algorithm suggested in [31].

8.3 Structure of Proposed Video Coder Framework

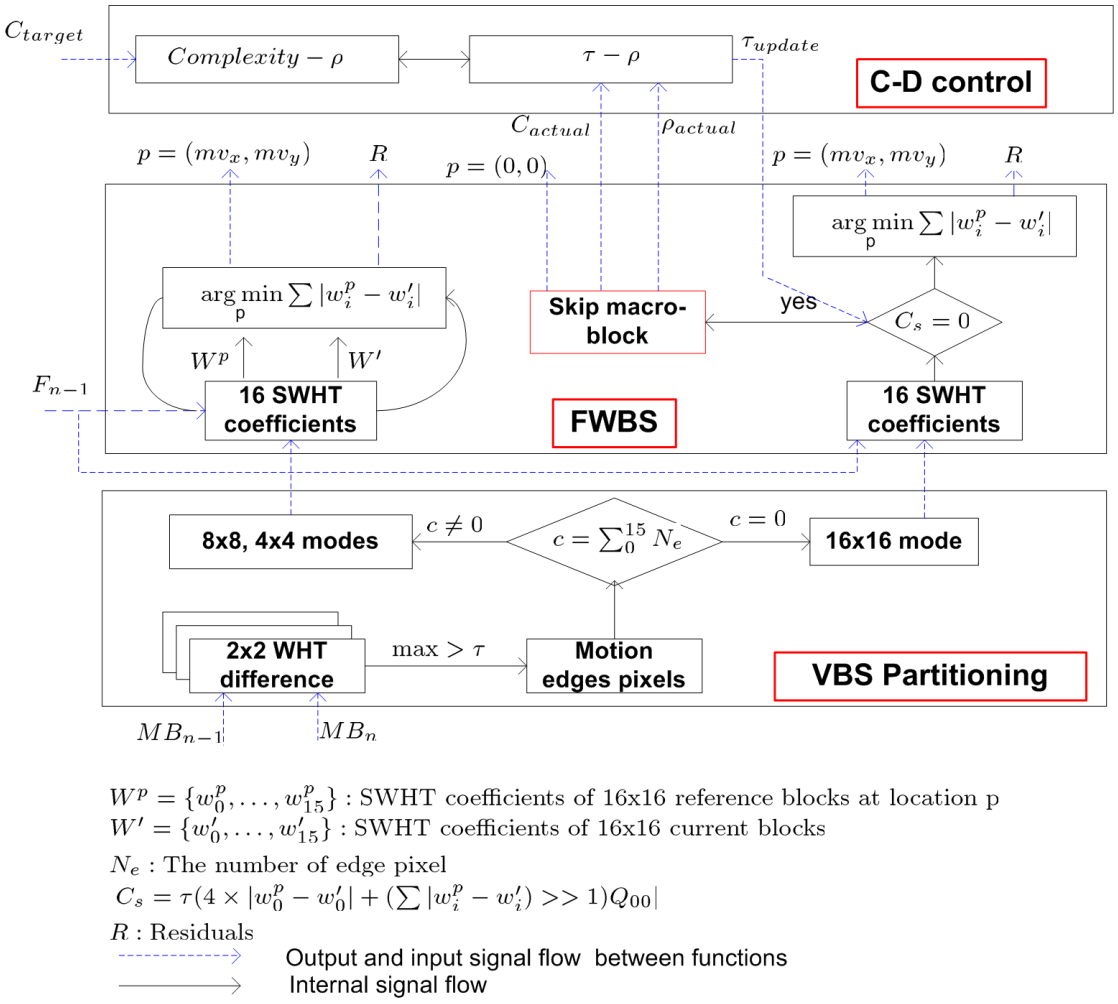


Figure 8.1: Overall structure of complexity adapted video coder framework;

As shown in Figure 8.1, the structure of a complexity adapted video coder consists of three parts; VBS partitioning, FWBS and controlling of C-D, where a skip MB detection algorithm is integrated into the FWBS block. In the VBS block, a 2×2 SWHT is performed to determine whether a block has a motion edge or represents a flat region. When the current MB to be encoded has no motion edges, this block is classified as 16×16 block size. Otherwise, the block is

further separated into several small block sizes such as 8×8 and 4×4 according to location of motion edge pixels. The FWBS block performs fast ME for 16×16, 8×8 and 4×4 blocks. This simple process is as follows; (1) Calculate to absolute difference between 16 lower frequency coefficients of the current block (W) and its corresponding reference block (W^p) in the previous frame at position p using Lemma 6.2 (see Chapter 6.4.1). (2) Find the minimum value at position p, then p becomes the motion vector (mv_x, mv_y). (3) p and residual (R) are transmitted to an entropy encoding block, which is outside the scope of this thesis. For a 16×16 block, FWBS is slightly modified by including a skip MB detection algorithm. If condition of skip MB detection (C_s)) is not zero, ME is performed as for other small blocks. Otherwise, a block is considered as a skip MB, which is the object of a C-D control block. Finally, Complexity-ρ and $\tau - \rho$ lines are used for complexity adaptation by controlling the threshold value (τ) of the skip ME detection algorithm.

8.4 Complexity Control Algorithm

In the remainder of this chapter, the complexity control algorithm is presented. The proposed algorithm consists of two parts; (1) Complexity-ρ model, (2) Complexity adaption algorithm. Details are discussed in the following sections.

8.4.1 C-D optimization using Lagrangian Multiplier

R-D theory has been widely used in video compression to obtain minimum bit rate at a given distortion constraint or vice versa. The Lagrangian multiplier method has been applied to H.264/AVC [36, 121, 126] to solve constrained optimization problems. Moreover, the Lagrangian multiplier method can be used for C-D based on complexity distortion theory [100] for both constrained and unconstrained optimization problems. In a constrained optimization problem, Problem 8.1 represents a general optimization procedure.

Problem 8.1. Given a set of coding parameters $P = \langle p_1, p_2, \ldots, p_n \rangle$, a sequence of macro blocks $\langle m_1, m_2, \ldots, m_n \rangle$, and a target complexity budget C, determine an assignment of coding parameters to each block that minimizes a distortion measure $D(P)$ using $C(P) \leq C_{target}$.

Problem 8.1 can be transformed to the following unconstrained optimization problem.

Problem 8.2. Given a set of coding parameters $P = \langle p_1, p_2, \ldots, p_n \rangle$, a sequence of macro blocks $\langle m_1, m_2, \ldots, m_n \rangle$, and a Lagrangian multiplier λ_P, determine an assignment of coding parameters to each block that minimizes the cost function $J(P) = D(P) + \lambda_P C(P)$.

In R-D optimization, QP is the main coding parameter used to solve the optimization problem. However, in C-D optimization, various coding parameters affect computational complexity. For example in H.264/AVC, complexity is greatly influenced by coding parameters such as search range, number of reference frames, presence of Hadamard transform, sub-pel accuracy ME and compensation and so on. Therefore, it is difficult to solve the C-D optimization using the Lagrangian multiplier method because it is very difficult to understand the effect of various coding parameters according to various sequences. Therefore, a more robust and simple C-D model is a mandatory tool to implement real-time applications running on power limited platforms.

8.4.2 Complexity-ρ Model

To create the complexity model based on the skip MB and zero motion detection algorithms mentioned above, we generate two curves according to their definitions and plot them for various situations. There are denoted as $\rho(\tau)$ and $C(\rho)$ which is computational complexity at given ρ, where τ is a control parameter of skip MB detection algorithm by multiplying the threshold value for deciding skip MBs. From Equation (7.28) (see Chapter 7.4), two control parameters are defined as:

$$\rho(\tau) = \frac{n_s}{n_t}, \quad Th = \tau \times \left(4W^{16}(0) + \left(\sum_{k=0}^{15} |W^{16}(k)| \right) >> 1 \right) Q_{00} \tag{8.1}$$

where n_s and n_t represent the number of skip and total MBs in a frame. In the following, let us consider video sequences "Foreman" and "Mother and Daughter" with CIF resolution. Figure 8.2 and Figure 8.3 show the relationship between ρ and τ, and complexity and ρ respectively. Each sampled picture is taken at every fourth frame. The sample pictures from "Foreman" and "Mother and Daughter" have different characteristics. For example, the "Foreman" sequence

has lots of motion with camera panning. On the contrary, the "Mother and Daughter" sequence has a lot of plain area. However, their characteristic curves for $\rho(\tau)$ represent very similar patterns, i.e. monotonic decrease over τ. The $C(\rho)$ curve shows almost the same pattern as well; inverse relationship of first order. From observations, the computational complexity ($C(\rho)$) is much more closely related to the percentage of skip MBs, ρ. For each sampled picture, $C(\rho)$ shows a straight line. When ρ is 1, $C(\rho)$ theoretically becomes zero because all MBs are chosen as skip, which leads to no processing of an encoder except for only copying the previous frame. Therefore, the line must pass through the point $(\rho, C(\rho)) = (1, 0)$. The following expression can be obtained from the above observations.

$$C(\rho) = k(1 - \rho) \tag{8.2}$$

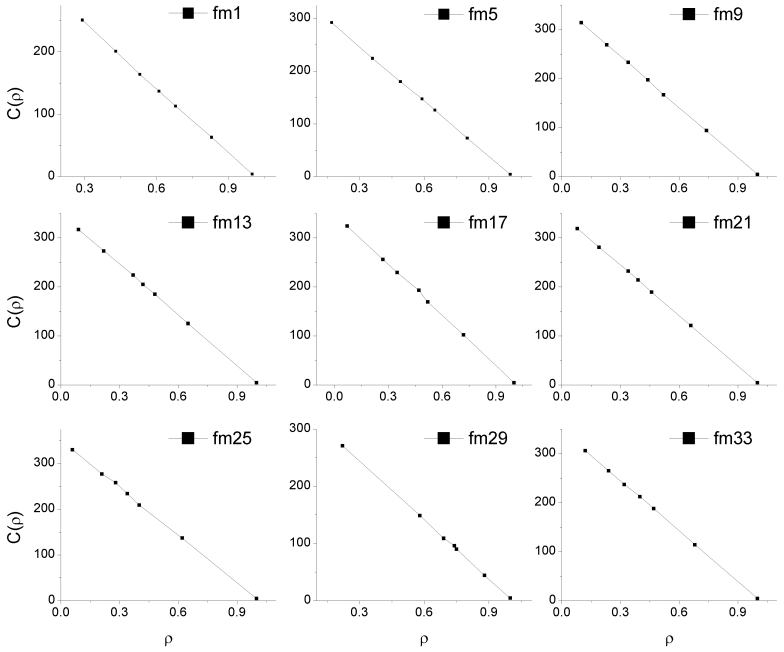
where k is a relative constant, which depends on picture characteristic as shown in Figure 8.2 and Figure 8.3. Equation (8.2) has been validated in our extensive simulation using various video sequences.

8.4.3 Complexity Adaptation Algorithm

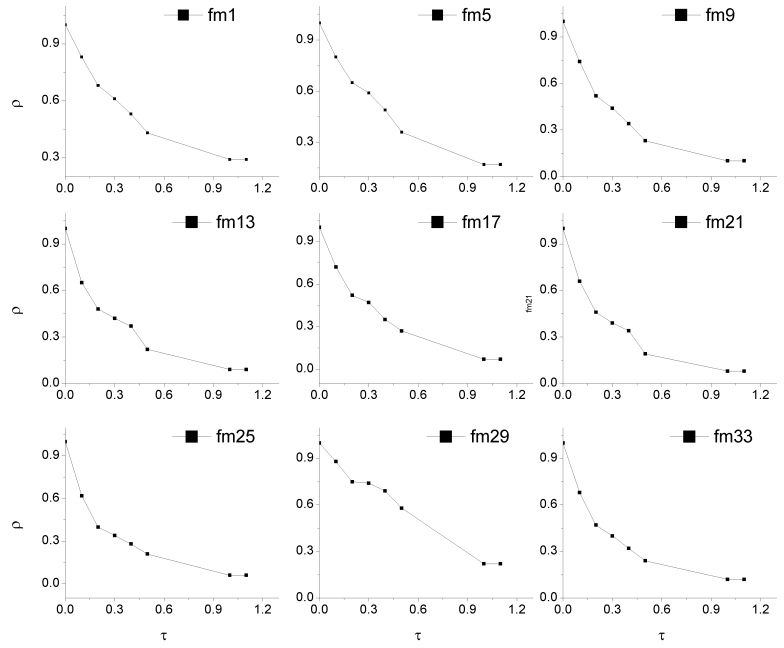
If we estimate the complexity of an initial P-frame $(\rho(1), C(\rho(1))$ at $\tau = 1$, and the other point for the complexity model is initially set to $(\rho, C(\rho)) = (1, 0)$. Then the complexity model denoted in Equation (8.2) becomes

$$C(\rho) = \left(\frac{C(\rho(1))}{1 - \rho(1)}\right)(1 - \rho). \tag{8.3}$$

Figure 8.4 shows an example of how to determine τ of the first P-frame. Firstly, $\tau_{initial} = 1$ is chosen as an initial step, which represents the maximum number of skip MBs without affecting the R-D performance. For example, when ρ is k after ME, DCT, and quantisation in a H.264/AVC, $\rho(1)$ has the same value as $\rho = k$ if the skip MBs are perfectly detected using skip algorithms. Therefore, ρ is a constant k in the case of τ greater than "1", so we only consider the range of $\tau \in [0, 1]$. Secondly, $C(p(1))$, which represents the maximum complexity C_{max}, is measured via encoding the first P-frame. Then we can draw the Complexity-ρ line using two points; $(\rho(1), C(\rho(1)))$, (1,0). Thirdly, the target (ρ_{target}) can be calculated corresponding to the target complexity (C_{target}) in the complexity-ρ plot. Finally, τ_{target} is also obtained in the ρ-τ plot. However, there remain

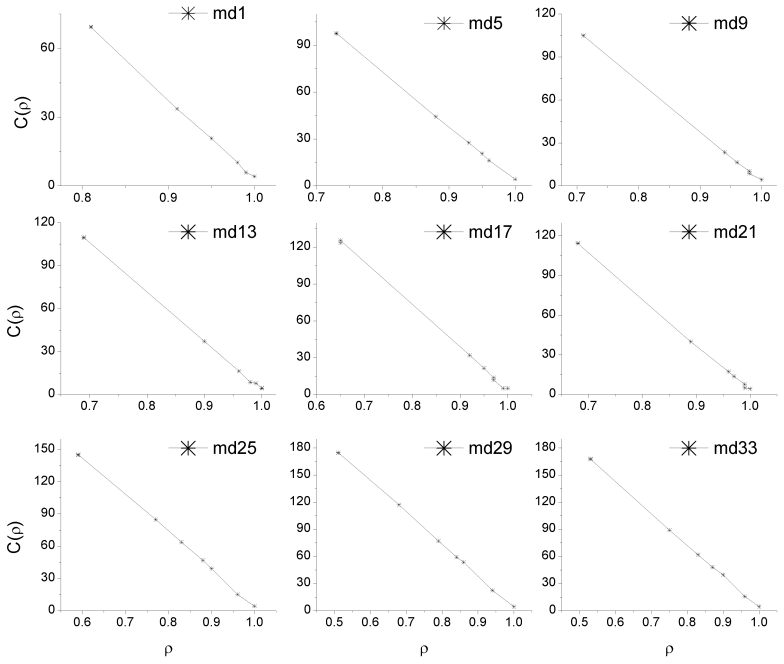


(a) Plot of $C(\rho)$ for the 9 sampled pictures from "Foreman." The x axis represents the percentage of skip macro-blocks ρ while the y axis represents the computational complexity $C(\rho)$.

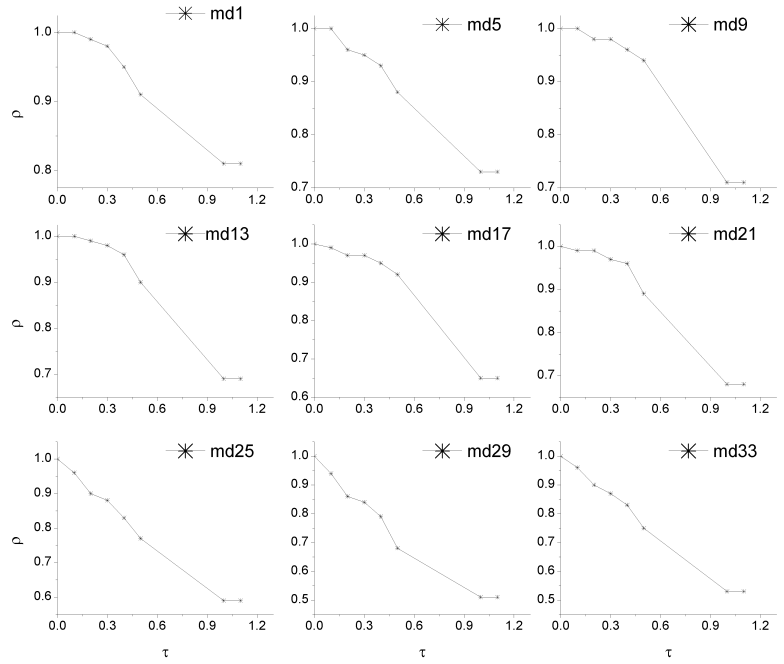


(b) Plot of $\rho(\tau)$ for the 9 sampled pictures from "Foreman." The x axis represents the threshold value of decision skip macro-blocks τ while the y axis represents the percentage of skip macro-blocks ρ.

Figure 8.2: Plot of $\rho(\tau)$ and $C(\rho)$ for "Foreman" at fixed $Qp = 30$.



(a) Plot of $C(\rho)$ for the 9 sampled pictures from "Mother and Daughter." The x axis represents the percentage of skip macro-blocks ρ while the y axis represents the computational complexity $C(\rho)$.



(b) Plot of $\rho(\tau)$ for the 9 sampled pictures from "Mother and Daughter." The x axis represents the threshold value of decision skip macro-blocks τ while the y axis represents the percentage of skip macro-blocks ρ.

Figure 8.3: Plot of $\rho(\tau)$ and $C(\rho)$ for "Mother and Daughter" at fixed $Qp = 30$.

some issues in obtaining τ_{target}; (1) we have no accurate relationship between τ and ρ, we only know it shows monotonic decrease and similar characteristics among frames in the same sequence as denoted in Figure 8.2 and Figure 8.3. (2) we need multi-path coding in order to obtain accurate τ-ρ plot, which is not realistic in complexity constrained applications such as real time on power limited platforms. Therefore, we present frame level complexity adaptation by updating parameters to solve the above problems.

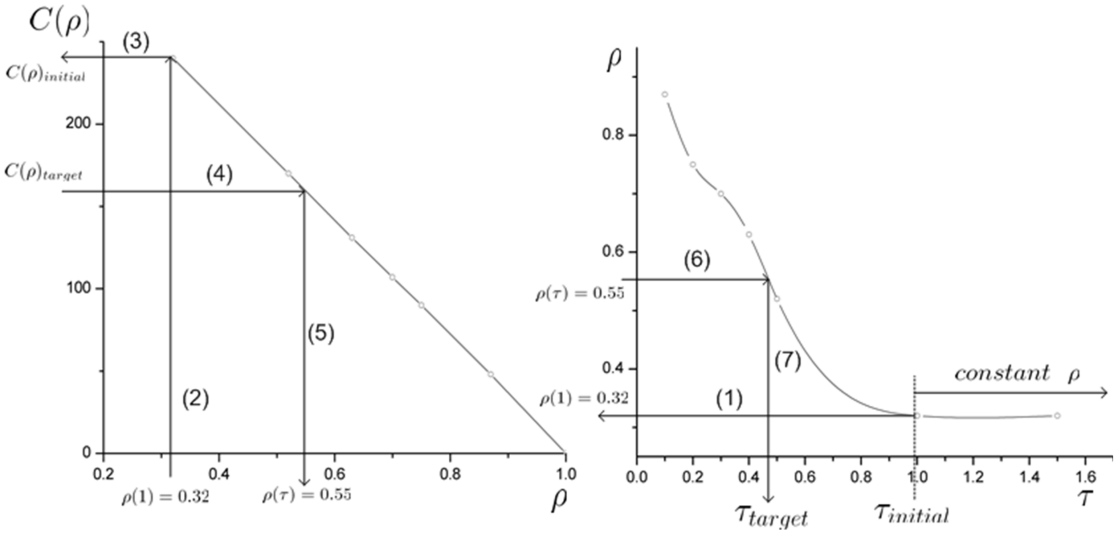


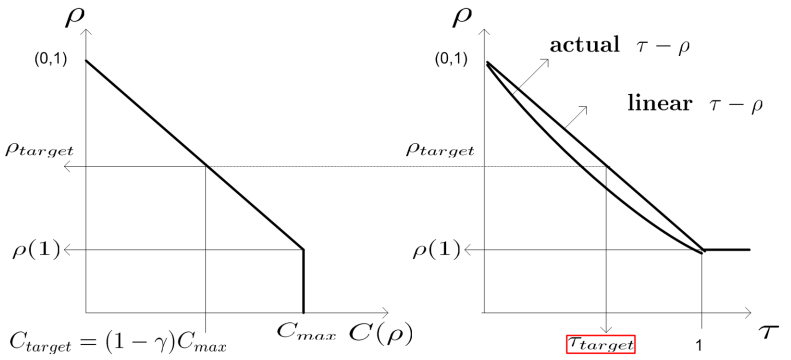
Figure 8.4: Example of estimating of τ_{target} in Foreman sequence using $C(\rho)$ and $\rho\tau$; (1) choose $\tau_{initial} = 1.0$ and find $\rho(1)$, which means a marginal ratio of skip MBs, therefore constant ρ when $\tau \geq 1$; (2)(3) obtain $C(\rho)_{initial}$ in Complexity-ρ plot. The straight line can be calculated from the two points $(\rho, C(\rho)) = \{(1,0), (\rho(1), C(\rho)_{initial})\}$ using Equation (8.3); (4)(5) calculate required $\rho(\tau)$ at given target complexity $C(\rho)_{target}$ using Complexity-ρ model. (6)(7) finally, τ_{target} can be used as a threshold for the decision of skip MBs in the current frame.

Frame Level Complexity Control Algorithm

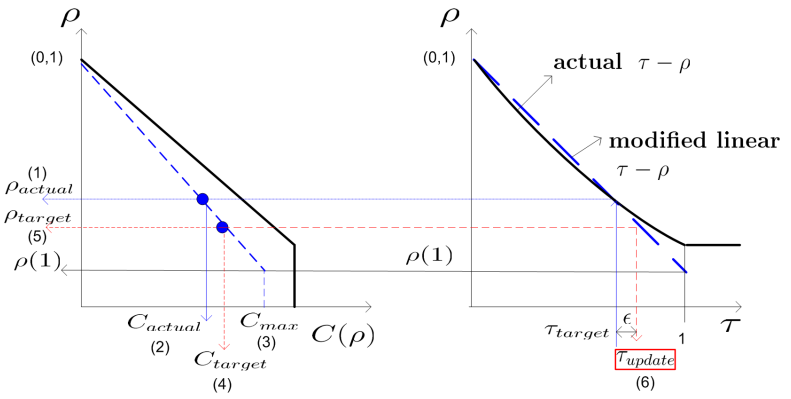
The complexity model of Equation (8.3) is used for the first P-frame. It must be noted that Equation (8.3) is based on the assumption that each frame has similar characteristics to nearby frames. Thus, scene changes may cause an error in the complexity-ρ model. Therefore, we need some assumptions to apply the complexity-ρ model as follows.

- P-frames between I-frames maintain the scene characteristics; i.e. keep the same shape of $\rho - \tau$, which can be validated by observing Figure 8.2 and Figure 8.3.
- The complexity of other functions such as skip MB detection and copy operation is negligible; the Complexity-ρ plot has a junction point (0,1) with the $C(\rho)$ axis in Complexity-ρ line.

Figure 8.5 shows the frame level complexity control algorithm graphically. Since only frame level complexity control method is used, several frames are needed to stabilize complexity adaption.



(a) Illustration of the complexity control on the first P-frame; $\rho(1)$ and C_{max} are obtained via marginally two point (0,1),(1,0). τ_{target} is calculated in linear $\tau - \rho$ line. Red box represents a passing parameter to the following P-frame



(b) Illustration of the complexity control on the following P-frame; the number indicates the order of obtaining parameters. τ_{update} is passed to the following P-frame

Figure 8.5: Graphical illustration of the frame level complexity control algorithm

We introduce the linearly fitted $\tau - \rho$ line using two points ((0,1), (1,ρ) as follows,

$$\rho = (\rho(1) - 1)\tau + 1. \tag{8.4}$$

An unknown actual $\tau - \rho$ line is also shown as a monotonically decreasing function. We consider the scenario that the user wants to control computational complexity by setting a reduction ratio (γ), given by

$$\gamma = \frac{C_{max} - C_{target}}{C_{max}}. \tag{8.5}$$

Figure 8.5(a) illustrates the complexity adaptation procedure for the first P-frame. $C(\rho)$ can be obtained using Equation (8.3). Moreover ρ_{target} is calculated via Complexity-ρ and expressed as follows:

$$\begin{aligned} \rho_{target} &= (\rho(1) - 1) \times \frac{C_{target}}{C_{max}} + 1 \\ &= (1 - \rho(1)) \times (\gamma - 1) + 1 \end{aligned} \tag{8.6}$$

τ_{target} is calculated using Equation (8.4), and passed through to the following P-frame as an index threshold value.

Figure 8.5(b) shows how to determine τ_{update} to correct the error caused by the unknown characteristic of $\rho - \tau$ line in the following P-frame. We obtain ρ_{actual} by encoding a P-frame with threshold τ_{target} received from the previous P-frame. The $\tau - \rho$ line is calculated using two points, $(\tau_{target}, \rho_{actual})$,(0,1). Equation (8.4) can be rewritten as follow:

$$\rho = \left(\frac{\rho_{actual} - 1}{\tau_{target}} \right) \tau + 1. \tag{8.7}$$

$\rho(1)$ can be obtained using Equation (8.7), which is given by

$$\rho(1) = \left(\frac{\rho_{actual} - 1}{\tau_{target}} \right) + 1. \tag{8.8}$$

From the Complexity-ρ line, C_{max} is calculated by finding the intersection point with $\rho(1)$ as follow:

$$C_{max} = \frac{C_{actual}}{1 - \rho_{actual}} (1 - \rho(1)). \tag{8.9}$$

Therefore, we calculate C_{target} using Equation (8.5) in the Complexity-ρ line denoted by

$$C_{target} = (1 - \gamma) C_{max}. \tag{8.10}$$

Moreover, ρ_{target} is also calculated in the Complexity-ρ line , which is given by

$$\rho_{target} = (\rho_{actual} - 1)\frac{C_{target}}{C_{actual}} + 1. \tag{8.11}$$

Finally, τ_{update} is calculated in the $\tau - \rho$ line using Equation (8.12), and passed to the following P-frame. This procedure repeats until an I-frame occurs.

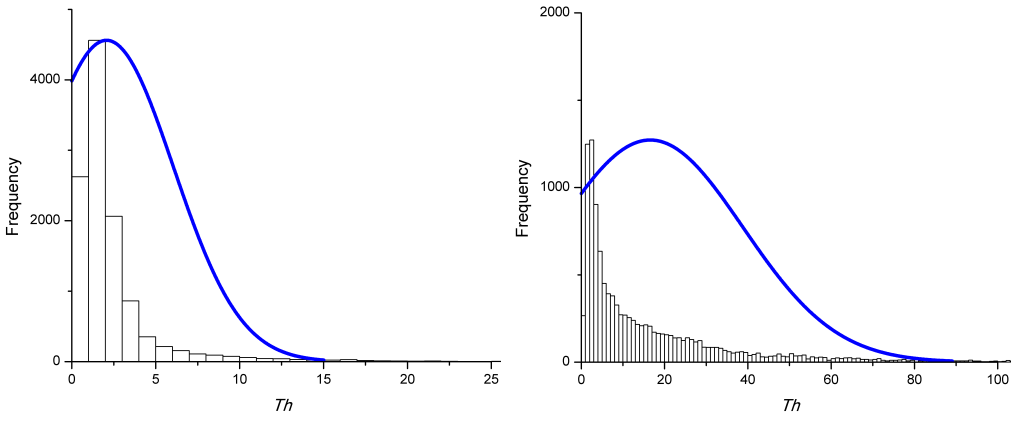
$$\begin{aligned} \tau_{update} &= \left(\frac{1 - \rho_{target}}{1 - \rho_{actual}}\right)\tau_{target} \\ &= \kappa\tau_{target} \end{aligned} \tag{8.12}$$

Where κ represents a correction constant.

8.4.4 Results

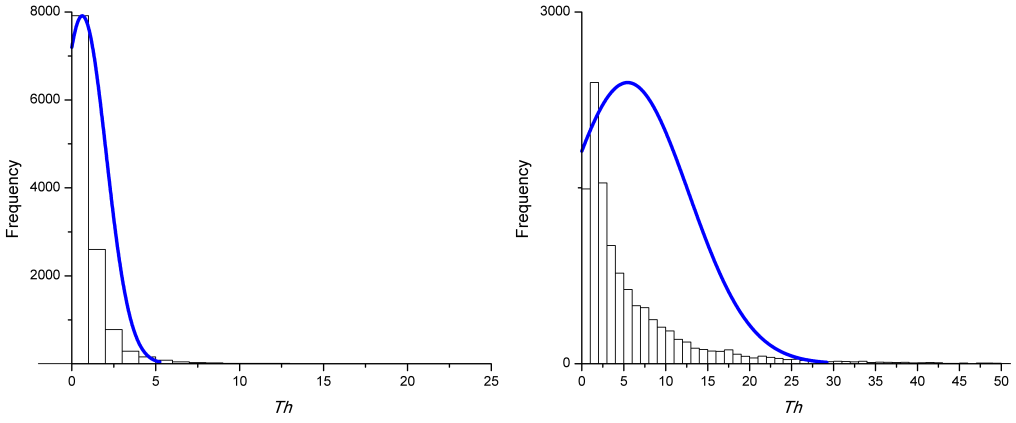
In order to understand the relationship between $\rho(1)$ and QP, a histogram of threshold defined in Equation (8.1) at $\tau = 1$ for Foreman and Mother and Daughter sequences as shown in Figure 8.6. As QP increases, the distribution of Th becomes narrow. This means that a small variation of Th results in large reduction of the computational complexity. Moreover, $\rho(1)$ is also increased, which lead to the limitation of a complexity control algorithm especially for Mother and Daughter sequence because the $\rho(1)$ is convergent to 1. This means that all blocks are skipped as shown in Figure 8.6(e). In the Foreman sequence, $\rho(1)$ does not reach 1 even for high QP. The small variation of Th results in not much affecting of the complexity control algorithm than Mother and Daughter case. In a conclusion, there is much room for complexity control in low QP, where the $\rho(1)$ is far from convergent value denoted as 1.

A number of test sequences from QCIF to SD are encoded using the proposed complexity control algorithm. Table 8.1 shows the test conditions. The sequences are coded at QPs (24 and 36) with complexity reductions from 0.1 (10%) to 0.5 (50%) of the JM modified via inclusion of VBS, FWBS, and skip MB. Table 8.2 shows the degradation caused by complexity reduction both in terms of PSNR and Bit-rate at a given reduction ration as denoted in Equation (8.5). Moreover, the achieved complexity reductions are also shown with γ'. The γ' shows some error at a given γ because the proposed method is only performed at the frame level. Thus, in order to adjust complexity with γ, the



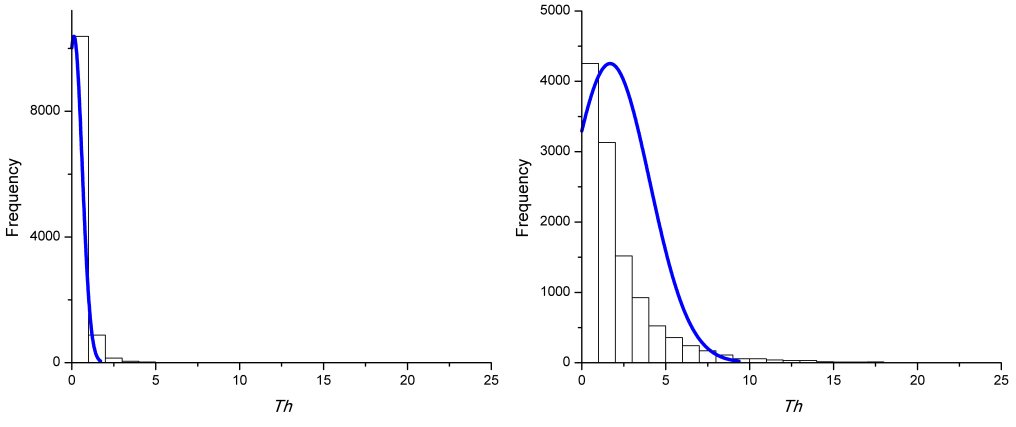
(a) $\rho(1) = 0.23$ at QP =20

(b) $\rho(1) = 0.02$ at QP=20



(c) $\rho(1) = 0.66$ at QP= 30

(d) $\rho(1) = 0.13$ at QP= 30



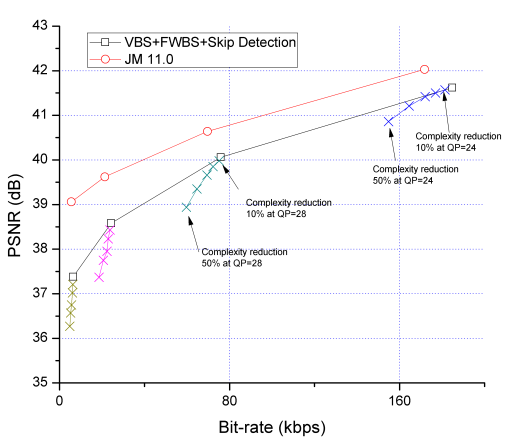
(e) $\rho(1) = 0.90$ at QP= 40

(f) $\rho(1) = 0.37$ at QP= 40

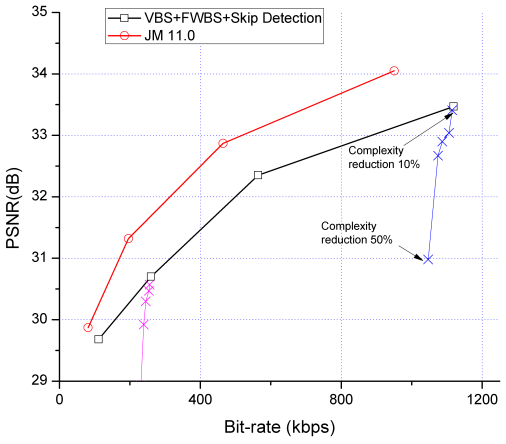
Figure 8.6: Histogram of Th value at $\tau = 1$ (see Equation (8.1)); (a)&(c)&(e) Mother and Daughter, (b)&(d)&(f) Foreman. Complexity control algorithm is limited at high QP for Mother and Daughter because $\rho(1)$ is convergent to 1, which means all blocks are skipped.

Table 8.1: Test conditions of the complexity adapted video encoder framework

Test Condition (JM 11.0)	
Sequences	All sequences mentioned in Section 3.2.1
GOP	IPPP structure, no B-frame
Evaluation	PSNR, Bit-rate
Test Platform	Intel dual core 3.0GHz, WinXP, Visual studio 6
Test Procedure	
	Encoding Parameters
VBS	16x16,8x8,4x4 mode, τ = 1.5QP (see Chapter 5.4.3)
Hadamard	On
Sub-pel ME	1/4-pel accuracy (replace integer-pel with FWBS)
Search Range	16
# reference frame	1
CAVLC / CABAC	CAVLC



(a) Mother and Daughter@352 × 288; R-D performance for all γ' for a given γ



(b) Foreman@352 × 288; R-D performance for all γ' for a given γ

Figure 8.7: R-D performance for all γ

complexity adaptation algorithm should consider the MB level. However, this introduces additional complexity because the adaptation procedures are performed at every MB. In complex motion sequences such as "Foreman", "Rush hour" and "Pedestrian", considerable PSNR degradation occurs at a high complexity reduction ratio such as 0.4 or 0.5. Those sequences have complex motions, which generates large coefficients values. When the complexity reduction is performed on those coefficients, lots of rounding errors occur, which represents considerable PSNR degradation. For $\Delta Bitrates$, as γ increase, required bit rates are decreased due to more MBs are decided as skip ones. Moreover, the degradation of the PSNR is more severe at low QP rather than in high QP due to rounding errors caused by setting a non skip MB to a skip MB by compulsion. On the

Table 8.2: Target and actual complexity reduction and R-D performance

Sequences	QP	$\gamma(\frac{C_{max}-C_{target}}{C_{max}})$	$\gamma'(\frac{C_{max}-C_{actual}}{C_{max}})$	$\Delta PSNR(dB)$	$\Delta Bitrates(\%)$
Foreman 176x144 30Hz	24	0.1	0.078	+0.02	-0.59
		0.2	0.145	-0.43	-2.90
		0.3	0.260	-1.40	-11.10
		0.4	0.373	-2.75	-18.86
		0.5	0.582	-6.74	-37.12
	36	0.1	0.105	-0.19	-7.61
		0.2	0.168	-0.42	-7.61
		0.3	0.248	-0.74	-8.18
		0.4	0.374	-1.18	-13.38
		0.5	0.527	-1.98	-20.95
Mother and Daughter 176x144 30Hz	24	0.1	0.148	-0.15	-1.38
		0.2	0.248	-0.27	-5.41
		0.3	0.356	-0.48	-6.54
		0.4	0.405	-0.64	-8.81
		0.5	0.477	-0.79	-12.14
	36	0.1	0.101	-0.10	-10.15
		0.2	0.170	-0.23	-20.31
		0.3	0.300	-0.25	-35.70
		0.4	0.391	-0.27	-43.69
		0.5	0.505	-0.31	-53.84
Foreman 352x288 30Hz	24	0.1	0.092	-0.06	-3.46
		0.2	0.188	-0.43	-7.26
		0.3	0.263	-0.57	-11.24
		0.4	0.367	-0.80	-16.42
		0.5	0.531	-2.49	-27.83
	36	0.1	0.101	-0.13	-3.51
		0.2	0.208	-0.23	-5.61
		0.3	0.292	-0.40	-7.02
		0.4	0.396	-0.78	-11.23
		0.5	0.534	-1.77	-14.39
Mother and Daughter 352x288 30Hz	24	0.1	0.161	-0.11	-2.36
		0.2	0.233	-0.22	-6.13
		0.3	0.339	-0.37	-9.43
		0.4	0.394	-0.49	-10.85
		0.5	0.457	-0.69	-15.09
	36	0.1	0.078	-0.07	-7.70
		0.2	0.237	-0.23	-18.31
		0.3	0.297	-0.29	-22.47
		0.4	0.358	-0.38	-27.47
		0.5	0.539	-0.54	-37.77
Pedestrian 720x576 25Hz	24	0.1	0.086	-0.15	-0.60
		0.2	0.137	-0.44	-2.24
		0.3	0.229	-1.44	-4.19
		0.4	0.381	-3.68	-6.46
		0.5	0.532	-6.21	-8.18
	36	0.1	0.057	-0.11	0.00
		0.2	0.204	-0.59	-1.17
		0.3	0.316	-1.33	-1.17
		0.4	0.381	-1.50	-1.17
		0.5	0.474	-2.07	-1.34
Rush Hour 720x576 25Hz	24	0.1	0.136	-0.21	-2.14
		0.2	0.194	-0.52	-4.80
		0.3	0.286	-0.81	-6.86
		0.4	0.406	-1.78	-12.40
		0.5	0.595	-4.54	-26.48
	36	0.1	0.095	-0.14	-0.72
		0.2	0.225	-0.39	-3.28
		0.3	0.305	-0.60	-5.47
		0.4	0.396	-0.92	-8.76
		0.5	0.526	-1.46	-11.68

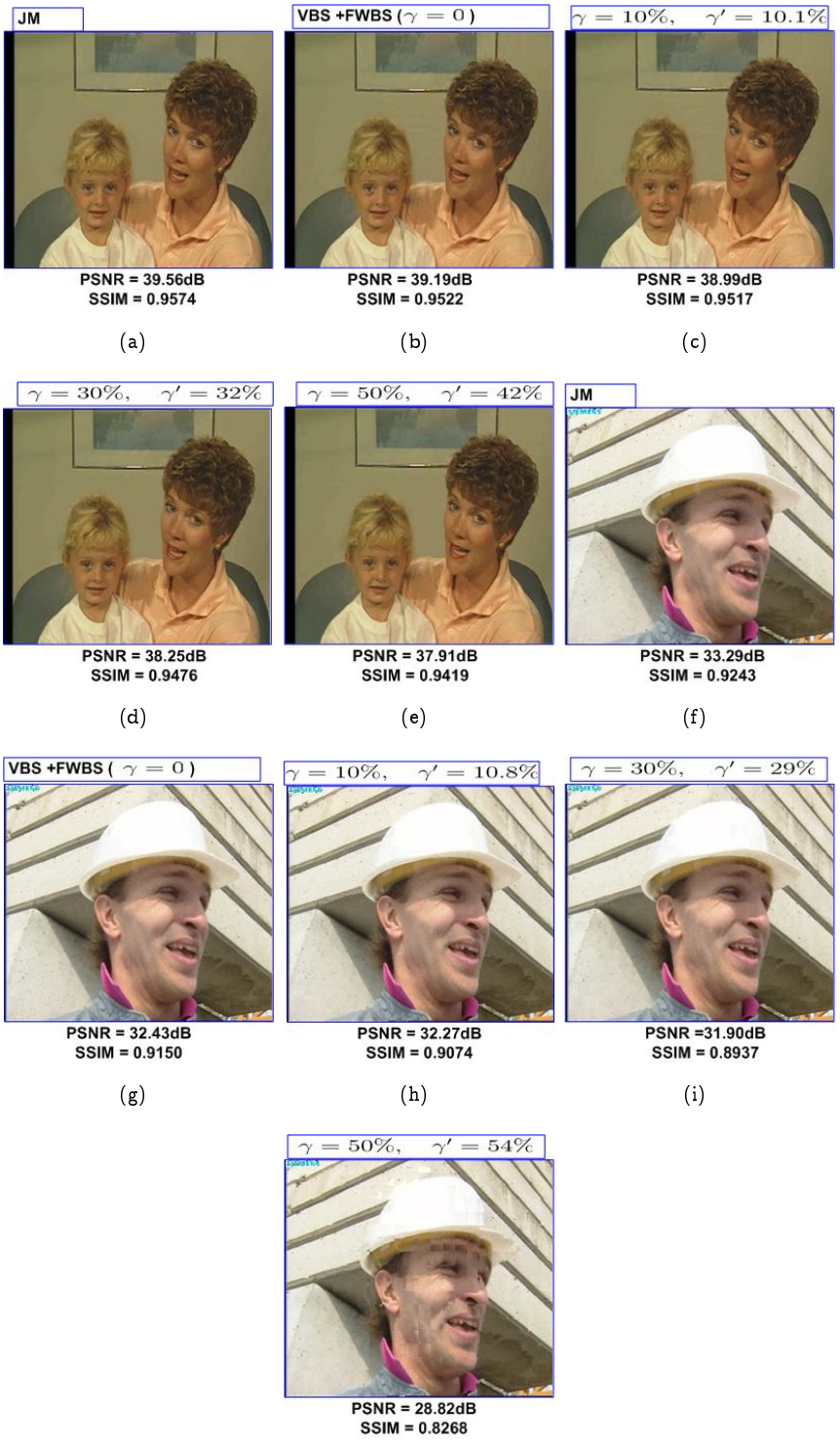
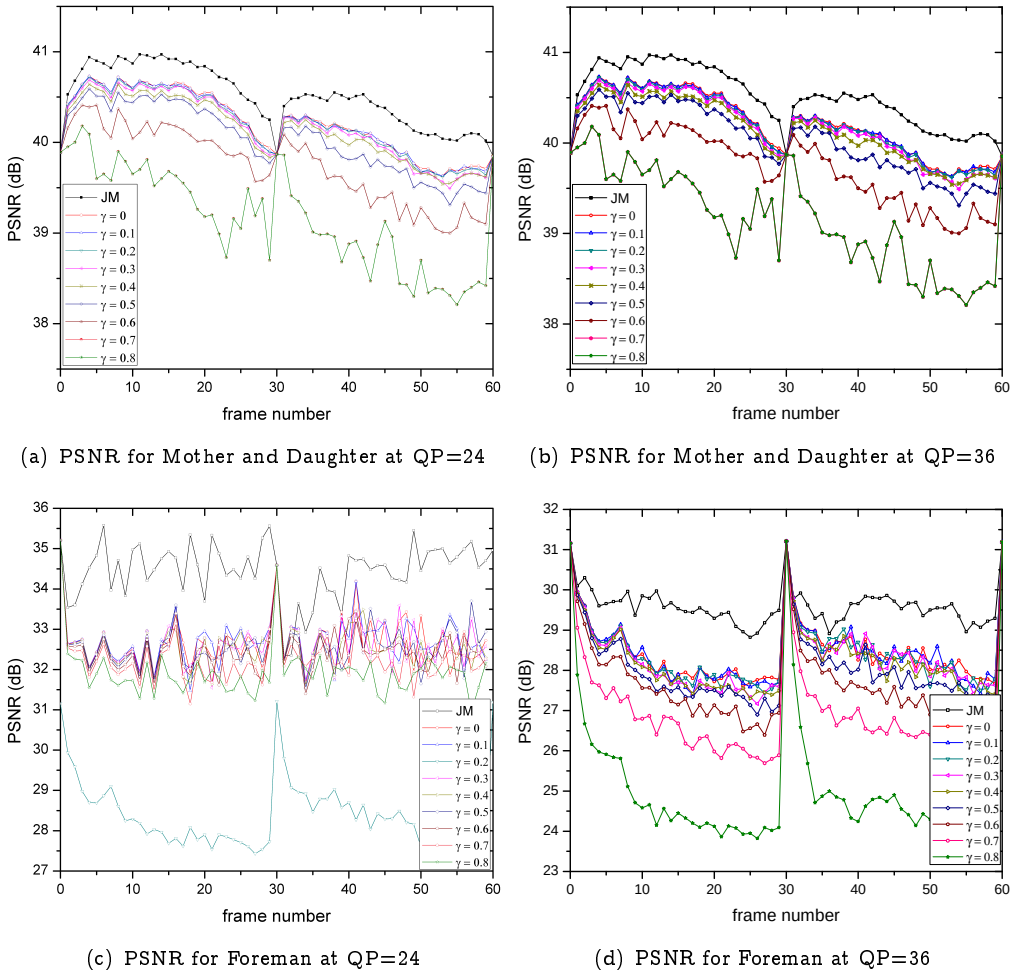


Figure 8.8: Visual comparison between the proposed algorithm and the JM; 10^{th} frame of the Mother and Daughter and the Foreman sequence; all tests are done at QP=26. γ is target complexity reduction denoted in Equation (8.5). γ' is actual complexity reduction



(a) PSNR for Mother and Daughter at QP=24

(b) PSNR for Mother and Daughter at QP=36

(c) PSNR for Foreman at QP=24

(d) PSNR for Foreman at QP=36

Figure 8.9: PSNR performance of the algorithm with the variation of γ

contrary, not much degradation of the PSNR is shown for "Mother and Daughter" sequences. Figure 8.7 shows R-D performance between the JM and the proposed approach at various γ. "Foreman" sequence shows a steep gradients of R-D compared to "Mother and Daughter" sequences. Therefore, more complexity reduction (γ) leads to poorer R-D performance for a complex motion sequence.

Figure 8.8 shows reconstructed frames from "Mother and Daughter" and "Foreman" sequence (frame 10^{th}, QP =26) coded using the JM and the reduced complexity encoder according to γ. In the "Mother and Daughter" sequence, no significant difference can be observed for all range of γ as depicted in Figure 8.8(a)-(e). Note that γ' represents the actual reduction ration of complexity. For "Foreman" sequence, no significant difference is observed until $\gamma = 0.3(30\%)$. However, at $\gamma = 0.5$ (50%), significant PSNR degradation does occur. The reconstructed frame contains a lots of blocking artifact in a complex motion area

(see Figure 8.8(j)). The high reduction of complexity leads to significant PSNR drop for a complex motion sequences. Therefore, the complexity reduction ratio (γ) is limited for those sequences in terms of R-D performance.

Figure 8.9 shows the PSNR performance on a frame-by-frame basis for the algorithm for a given complexity reduction ratio (γ), for the "Foreman" and "Mother and Daughter" sequences encoded with QP values of 24 and 36. The graph shows a small PSNR drop if γ is less than 0.5. This small PSNR drop could be attributed to the algorithm incorrectly skipping some MB that should have been coded. Moreover, a large PSNR drop occur if γ is more than 0.5 due to rounding error for high frequency AC coefficients (there occur in MB containing edges of moving objects) using a large threshold.

Algorithm 1 Frame level complexity adaption

Require: F_i (i^{th} frame s.t $i \geq 0$), C_{target},γ, N is I-frame interval.
Ensure: τ_{update}
  if $F_i\%N = 1$ **then**
    $\tau \leftarrow 1$
    Obtain Complexity-ρ line using $(\rho(1), C(\rho(1))), (1, 0)$
    Calculate C_{target} using Equation (8.5)
    Calculate ρ_{target} and τ_{target} using Equation (8.4)
    $\tau_{update} \leftarrow \tau_{target}$
  else
    repeat
      Obtain ρ_{actual} and C_{actual} at given τ_{update}
      Obtain modified Complexity-ρ line using $(C_{actual}, \rho_{actual})$,(0,1)
      Find $\rho(1)$ in $\tau - \rho$ line
      Calculate C_{max}
      Calculate C_{target}
      Obtain τ_{update}
    until End of P-Frame
  end if

8.5 Discussion

This chapter investigates a complexity adaptation algorithm for an H.264/AVC encoder. The algorithm controls the number of skip MBs before performing ME. MBs predicted as "skipped" are not processed further, saving all further computation. The frame level complexity control algorithm is also presented by introducing the Complexity-ρ model.

Algorithm 1 provides a summary of the frame level complexity control algorithm. The proposed algorithm focuses on real-time operation on power limited platform, where some restrictions should be considered.

- Complexity adaptation without feedback information, which means that only one-path encoding is allowed.
- The traditional RDO requires huge complexity, so it is difficult to use in complexity constrained platforms.

In order to address these restrictions, the proposed algorithm uses single-path based complexity adaption, ME based on VBS by detecting edges, which gives good R-D performance whilst not using RDO in ME and mode decision in H.264/AVC.

—The magic of first love is our ignorance that it can ever end.

Benjamin Disraeli

9 Discussion and Conclusion

9.1 Introduction

This chapter presents conclusions and future work related to the research carried out in this thesis. The algorithms and experimental results are critically reviewed in Section 9.2. The main contributions of this research are summarized in Section 9.3. Possible directions for further research in relation to the main findings are also indicated in Section 9.4.

9.2 Thesis Review

The aim of this research has been to develop novel algorithms to adapt the computational complexity of an encoder so that the available processing resources are used efficiently or user requirements in terms of complexity are satisfied in order to maximize video quality. The significant contributions of this research can be classified into four algorithms and they have been presented in four chapters.

- Chapter 5 $\longrightarrow$ VBS partitioning algorithm based on motion edge detection.
- Chapter 6 $\longrightarrow$ Fast ME algorithm in the SWHT domain called FWBS.
- Chapter 7 $\longrightarrow$ A skip MB detection algorithm in conjunction with FWBS.
- Chapter 8 $\longrightarrow$ A complexity adaptation framework presented by complexity-ρ modeling.

Moreover, the fundamentals of a digital video system and necessary background for the research reported are overviewed in three chapters.

- Chapter 2 $\longrightarrow$ Overview of digital video representation.
- Chapter 3 $\longrightarrow$ Experimental method, test sequences, and quality measure metrics are overviewed in order to give a clear understanding of the high level system architecture used, and how performance results were compared.
- Chapter 4 $\longrightarrow$ Simple testing of H.264/AVC using JM reference software to find which encoding parameters have most effect on R-D and C-D performance.

Chapter 1 presents an introduction to the thesis, including a description of the motivation of the research, the problem statement of complexity adaptation in a video encoder, and a summary of research challenges. It also presents the objectives of the research, and a brief description of the key contributions arising from the research.

In Chapter 2, a review of digital video representation is presented. The digital video system described in this chapter reveals that there are two main functional blocks; digital video representation and block based video coding. In digital video representation, captured analog image pixels are converted to digital for storage, compression, and transmission. Prediction, Transform, and Entropy coding play a major role in block based video compression. Their concepts and principles are also described in this chapter.

In Chapter 3, test sequences are selected and discussed in terms of their usefulness for video coding research. Moreover, the most common used objective video quality metrics are briefly reviewed. The test procedure of this research is presented, where all the proposed low complexity algorithms are integrated in H.264/AVC and experimental results are compared with the JM reference software.

The encoding parameters of H.264/AVC that give a bit rate saving of about 50% over previous standards are reviewed and investigated in terms of computational complexity in Chapter 4. VBS, sub-pel accuracy motion vector resolution

ME, and CABAC have an influence on enhancing R-D performance. On the contrary, search range, multiple reference frames, and existence of the WHT do not give much benefit considering both R-D and C-D performance. However, I would like to note that these results are not definitive due to lack of test sequences, thus, different results may be apparent for some other sequences.

In Chapter 5, a VBS partitioning algorithm based on motion edge detection is proposed. The binary edge map is obtained via low computational complexity edge detection using the WHT for 2×2 partitioned blocks. Two findings are derived from mathematical analysis. Firstly, inter prediction errors are mainly caused by the spatial gradients and their motion vectors. Therefore, the prediction error near motion edges becomes serious. Secondly, the threshold value in detecting motion edges is linearly related to QP, which gives a way to predict the VBS partitioning in order not to perform compression at given QP.

A fast ME algorithm called FWBS based on the WHT is presented in Chapter 6. The idea of this algorithm is based on a fundamental property of DTs, i.e., energy compactness. Two lemmas are proved in order to use basic tools of fast ME. One is that the RSSD in the pixel domain is bounded to the SAD in the transform domain. Therefore, complexity saving is achieved via calculating SAD on only a few transformed coefficients. For example, let block size for ME be a 16×16 block, only 16 low frequency coefficients occupy more than 95% of the energy of all blocks. Therefore, SAD is performed not for 16×16 whole block but for 16 coefficients. The other is that the relationship between a block and its sub-blocks in the SWHT is used for calculating 16 lower frequency coefficients using neighbouring blocks' coefficients. This has several advantages; (1) it requires less memory space, (2) the coefficients of the desired area can be obtained individually, which give a complexity saving.

In Chapter 7, a skip MB detection algorithm is proposed. A simple transform called the S-transform is introduced to find the relationship between ICT and SWHT. Moreover, the coefficients of the SWHT are bounded as the average maximum pixel values (called limitation of dynamic range) if all pixel values are positive. However, the motion compensated residuals could be negative values. Therefore, the statistical approach suggested in [18] is used in order to make all residues positive. Evenly adding or summing certain values to a block's pixels does not affect the variation of AC coefficients in the transform domain. Only the DC coefficient is changed. Therefore, we are justified in classifying this as a

skip MB if the compensated DC coefficient is zero after quantisation. The skip MB detection algorithm plays an important role in setting up the complexity adaptation framework described in Chapter 8.

Finally, a complexity adaptation video encoder framework is proposed in Chapter 8. The Complexity-ρ model is presented by observing the relationship between the threshold value of skip MB and the ratio of skip MB out of all candidate MBs. From simulation, the relationship between complexity and ρ can be linearly modeled with a first order curve, $C(\rho) = k(1-\rho)$. Moreover, an automatic complexity adaptation algorithm based on the complexity-ρ model, where the user can define the required level of complexity is presented. The results show that the complexity of an encoder is successfully adapted to user defined complexity although this is based on the hypothesis that adjacent frames have similar characteristics in terms of $\rho - \tau$ shape and the processing power of skip MB for memory copy is negligible.

9.3 Research Contributions

This thesis makes a number of research contributions related to complexity adaptation in video encoders. Novel algorithms were developed where building up the complexity adaptation framework. Key contributions of this research to the advancement of video coding or other applications can be summarized as follows:

- The development of a low computational complexity VBS algorithm: This algorithm can be extended to many fields such that (1) low complexity edge detection [46], (2) moving region segmentation [21], (3) shot boundary detection, and (4) video compression especially using sub-partition blocks such as the H.264/AVC.
- Fast ME algorithm (FWBS): This algorithm can be adopted to any video coding standard. H.264/AVC optionally support SATD. In this case, it gives benefits for complexity issues because no other processing is needed to find motion vectors. Moreover, it could be used in image matching especially low computational cost template matching.

- Skip MB detection algorithm: This algorithm can be applied to video coding standards in conjunction with FWBS. This algorithm is more effective at high QP values because the number of skip MB increases as QP rises.
- Complexity adaptation framework: The framework is very useful for applications running on power limited platforms such as CSNs, surveillance, and mobile video applications.

9.4 Challenges and Future Work

The algorithms developed during this research were summarized and critically evaluated in the previous section. This section presents challenges and directions for further research.

1. The proposed algorithms are integrated into the JM. However, its use has been limited because the JM is not an optimized version of H.264/AVC. Therefore, an optimized version of an encoder is necessary in order to apply the proposed algorithms in a real time encoder working on a power limited platform.
2. Only integer-pel accuracy ME is presented in this thesis. Sub-pel accuracy ME should be considered in further research. Interpolation filtering algorithms in the transform domain are one candidate to obtain sub-pel accuracy.
3. The complexity adaptation uses a frame level control algorithm, which requires several frames to reach the defined complexity. Therefore, more robust control algorithms such as MB level complexity control should be investigated.
4. In this research, baseline profile in H.264/AVC is considered, thus only P-frames are the main interesting of the complexity adaptation framework. In future research, we would like to turn to the main or high profile to consider B-frames.

A

Coefficient Relationship of sequency ordered Walsh Hadamard Transform (SWHT) between a block and its sub-blocks

A.1 One dimensional SWHT block and its sub-blocks

A radix-N point SWHT of signal $x(n)$, $n = 0, 1 \ldots N-1$, where N is power of 2, is defined as [88]

$$X(k) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n) \prod_{i=0}^{N-1} (-1)^{p_i(n) \times d_i(k)} \tag{A.1}$$

where $p_i(n)$ is the binary representative form of n, $d_i(k)$ can be defined followed by taking binary to gray-code (Equation (A.3)) and bit reversal conversion (Equation (A.2)) of the binary representation of k as follows:

$$\begin{aligned} n &= (p_{n-1} p_{n-2} \ldots p_0)_2 = \sum_{i=0}^{n-1} p_i 2^i \\ k &= (b_{n-1} b_{n-2} \ldots b_0)_2 = \sum_{i=0}^{n-1} b_i 2^i \end{aligned} \tag{A.2}$$

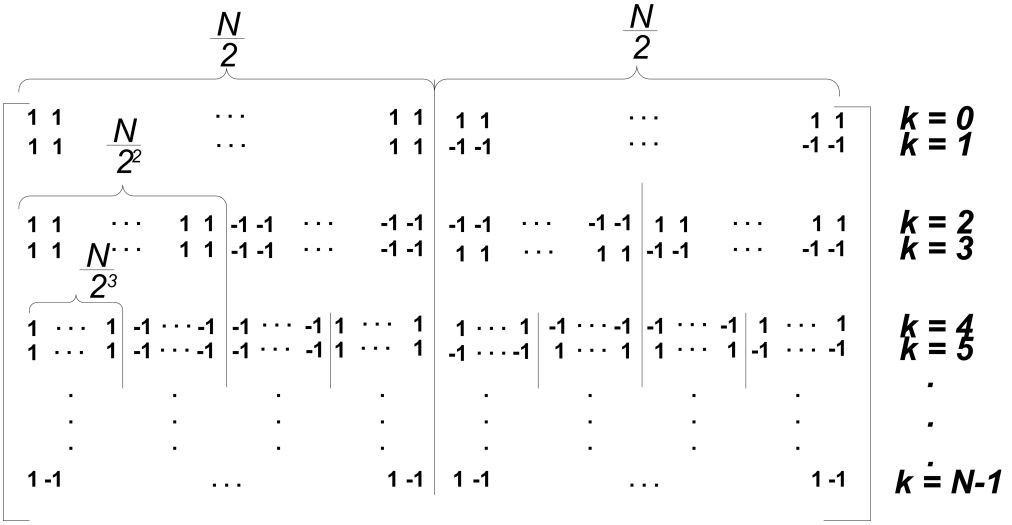
$$\begin{aligned} d_0 &= b_{n-1} \\ d_1 &= b_{n-1} + b_{n-2} \\ \vdots &= \vdots \\ d_{n-1} &= b_1 + b_0. \end{aligned} \tag{A.3}$$

The widely-used matrix representation of a signal transform is given as

$$X_N = T_N \mathbf{x},\ T_N = \begin{bmatrix} a_{0,0} & \cdots & a_{N,0} \\ \vdots & \ddots & \vdots \\ a_{0,N} & \cdots & a_{N,N} \end{bmatrix}. \tag{A.4}$$

When the vector forms of the signal $\mathbf{x} = \{x(0), \ldots, x(N-1)\}^T$ and transformed signal $X_N = \{X(0), \ldots, X(N-1)\}^T$ are used and the components of the $N \times N$ SWHT matrix T_N are obtained shown as in Equation (A.5) by observation of Equation (A.1), Equation (A.2), and Equation (A.3).

From Equation (A.3), every two bits of d_i have been repeated by increasing $k = k + 4$ since it changes two successive bits of b_i. We divide SWHT coefficients of X_N for a multiple of 4s, that is SWHT coefficients at $k = 4m$, $k = 4m + 1$, $k = 4m + 2$, and $k = 4m + 3$, where $m = 0, 1, \cdots, \frac{N}{4} - 1$. The $N \times N$ transform matrix is divided by its two sub matrices of $\frac{N}{2} \times \frac{N}{2}$ and the relationship between them can be observed as shown in Equation (A.6). where i, j represents row and



$$= \sqrt{N} T_N^{i,j} \tag{A.5}$$

column components of matrix $T_N^{i,j}$.

$$\begin{aligned}
\sqrt{2}T_N^{i,4m} &= \begin{cases} T_{N/2}^{i,2m} & i \in [0, N/2-1] \\ T_{N/2}^{i,2m} & i \in [N/2, N-1] \end{cases} \\
\sqrt{2}T_N^{i,4m+1} &= \begin{cases} T_{N/2}^{i,2m} & i \in [0, N/2-1] \\ -T_{N/2}^{i,2m} & i \in [N/2, N-1] \end{cases} \\
\sqrt{2}T_N^{i,4m+2} &= \begin{cases} T_{N/2}^{i,2m+1} & i \in [0, N/2-1] \\ -T_{N/2}^{i,2m+1} & i \in [N/2, N-1] \end{cases} \\
\sqrt{2}T_N^{i,4m+3} &= \begin{cases} T_{N/2}^{i,2m+1} & i \in [0, N/2-1] \\ T_{N/2}^{i,2m+1} & i \in [N/2, N-1] \end{cases}
\end{aligned} \tag{A.6}$$

Using the above results, we can derive the relationship of a block and its sub-blocks by introducing natural ordered Hadamard Transform [27]. The 2×2 Hadamard (H_2) and the sequency ordered Walsh Hadamard matrix (T_2) have the same definition given by

$$T_2 = H_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{A.7}$$

The natural order Hadamard transform is obtained by successive Kronecker multiplication ($\otimes$) of H_2 as follows:

$$\begin{aligned}
H_N &= H_2 \otimes H_{N/2} = H_{N/2} \otimes H_2 \\
&= \begin{bmatrix} H_{N/2} & H_{N/2} \\ H_{N/2} & -H_{N/2} \end{bmatrix} \\
&= \begin{bmatrix} H_{N/2} & \bar{0} \\ \bar{0} & H_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix}.
\end{aligned} \tag{A.8}$$

where $I_{N/2}$ is a $N/2 \times N/2$ identity matrix, $\bar{0}$ represents a zero matrix. The SWHT matrix of order N can be obtained by reordering sequency components of H_N. To convert a given sequency number of H_N into the corresponding index number of T_N, the permutation matrix is introduced based on the observation of the Equation (A.8), the SWHT transformation matrix T_N can be decomposed into combinations of sub matrix $T_{N/2}$ as

$$T_N = P_N \begin{bmatrix} T_{N/2} & \bar{0} \\ \bar{0} & T_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix} \tag{A.9}$$

where P_N is a permutation matrix of $N \times N$. N-point SWHT is decomposed into two $N/2$-point SWHTs. It is clear how to obtain the N-point SWHT, X_N, of $\mathbf{x}$ directly from the $N/2$-point SWHT, $\mathbf{X}_1 = \{X_1(0), \cdots, X_1(N/2-1)\}$ and $\mathbf{X}_2 = \{X_2(0), \cdots, X_2(N/2-1)\}$, of two sub-blocks; $\mathbf{x}_1$ and $\mathbf{x}_2$, where $\mathbf{x}_1 = \{x(0), \cdots, x(N/2-1)\}$ and $\mathbf{x}_2 = \{x(N/2), \cdots, x(N-1)\}$. After considering scalar scaling $\sqrt{N}$ in Equation (A.5), X_N can be obtained as follows;

$$\begin{aligned} X_N &= T_N \mathbf{x} \\ &= P_N \begin{bmatrix} T_{N/2} & \bar{0} \\ \bar{0} & T_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} \\ &= P_N \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{X}_1 + \mathbf{X}_2 \\ \mathbf{X}_1 - \mathbf{X}_2 \end{bmatrix} \\ &= P_N \left(\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \end{bmatrix} \right) \\ &= P_N \left(T_2 \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \end{bmatrix} \right) \\ &= P_N Q_N \end{aligned} \tag{A.10}$$

where Q_N is 2-point SWHT coefficients obtained with the two SWHT coefficients of sub-blocks. Let R_N perform reordering of a column vector by interleaving points from the first and second halves defined as

$$R_N = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 & 0 \cdots & 0 \\ 0 & 0 & \cdots & 0 & 1 & 0 \cdots & 0 \\ 0 & 1 & \cdots & 0 & 0 & 0 \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & 1 \cdots & 0 \\ & & \cdots & & & \cdots & \\ 0 & 0 & \cdots & 1 & 0 & 0 \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 \cdots & 1 \end{bmatrix} \tag{A.11}$$

P_N can be obtained using Equation (A.11) by order changing between the $(4m+2)^{th}$ and $(4m+3)^{th}$ column vector of R_N observing Equation (A.6)'s relationship

between a block and its sub-blocks matrix. Thus P_N can be seen that

$$P_N = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 & 0 \cdots & 0 \\ 0 & 0 & \cdots & 0 & 1 & 0 \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & 1 \cdots & 0 \\ 0 & 1 & \cdots & 0 & 0 & 0 \cdots & 0 \\ & & \cdots & & & \cdots & \\ 0 & 0 & \cdots & 0 & 0 & 0 \cdots & 1 \\ 0 & 0 & \cdots & 1 & 0 & 0 \cdots & 0 \end{bmatrix} \tag{A.12}$$

P_N acts as a interleaving two sub-blocks components, and reordering the SWHT coefficients at $4m$, $4m+1$, $4m+2$ and $4m+3$. Let reordering vectors of Q_N be $\overline{Q_N} = \{\overline{X}_1(0), \overline{X}_2(0), \ldots, \overline{X}_1(N/2-1), \overline{X}_2(N/2-1)\}^T$ depicted in Figure A.1, Equation (A.10) is simplified to

$$\begin{aligned} X_N &= P_N Q_N \\ &= (I_{N/2^2} \otimes S_2)\overline{Q_N} \\ , S_2 &= \begin{bmatrix} I_2 & \overline{0} \\ \overline{0} & \overline{I_2} \end{bmatrix}, \quad \overline{I_2} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \end{aligned} \tag{A.13}$$

where $\overline{0}$ is zero matrix, I_2 is a 2x2 identity matrix.

Figure A.1 illustrates a graphical relationship between N block and its $N/2$ sub-blocks denoted in Equation (A.13). $X(4m)$, $X(4m+1)$ are obtained the 2-point SWHT (T_2) between even number position of $\mathbf{X}_1$ and $\mathbf{X}_2$. On the contrary, $X(4m+2)$, $X(4m+3)$ are also obtained by taking 2-point SWHT using odd number position of $\mathbf{X}_1$ and $\mathbf{X}_2$ followed by 2×2 reflection matrix ($\overline{I}_2$), where $m = 0, 1, \cdots, N/4-1$. Equation (A.13) could be extended to arbitrary size of sub-blocks (power of 2) by recursion as follows;

$$\begin{aligned} X_N &= (I_{N/2^{i+1}} \otimes S_{2^i})\overline{Q_{N/2^i}} \\ , S_{2^i} &= \begin{bmatrix} I_{2^i} & \overline{0} \\ \overline{0} & \overline{I_{2^i}} \end{bmatrix} \end{aligned} \tag{A.14}$$

where $\mathbf{X}$ is divided by 2^i sub-blocks and $i = 1, \cdots, \log_2(N) - 1$. And $\overline{Q_{N/2^i}}$ is a interleaved 2^i SWHT of transformed data of sub-blocks $(X_1, X_2, \cdots, X_{2^i})$.

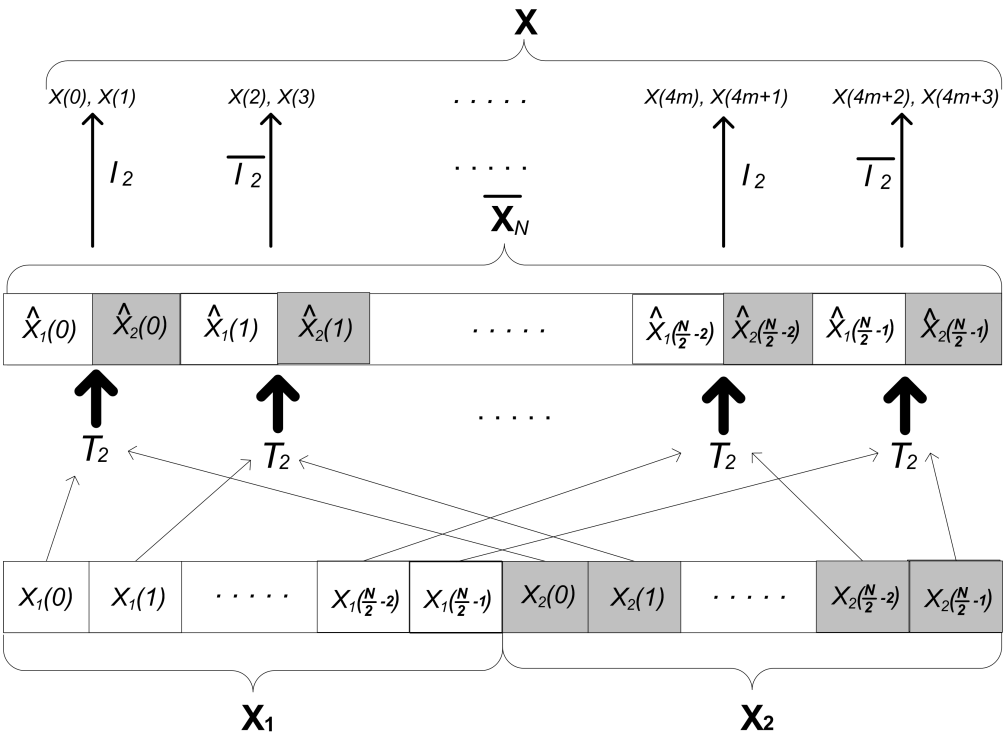


Figure A.1: Graphical representation of the relationship between N-point SWHT ($\mathbf{X}$) and $N/2$-point sub-blocks' SWHT ($\mathbf{X}_1, \mathbf{X}_2$)

A.2 Two dimensional SWHT block and its sub-blocks

The relationship between a 2-D SWHT block, X_{NN}, and its four sub-blocks can be determined through a similar method so that given in Section A.1. The direct relationship is given as

$$X_{NN} = T_N \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 \\ \mathbf{x}_3 & \mathbf{x}_4 \end{bmatrix} T_N^T \tag{A.15}$$

where $\mathbf{x}_i = [x_i(0), \ldots, x_i(N/2-1)]^T$, i represents the location of sub-blocks from raster scan order. Using Equation (A.9), we can observe that Equation (A.15) holds following;

$$\begin{aligned} X_{NN} =& P_N \begin{bmatrix} T_{N/2} & \bar{0} \\ \bar{0} & T_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 \\ \mathbf{x}_3 & \mathbf{x}_4 \end{bmatrix} \\ & \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix}^T \begin{bmatrix} T_{N/2} & \bar{0} \\ \bar{0} & T_{N/2} \end{bmatrix}^T P_N^T \\ =& \frac{1}{2} P_N Q_{NN} (P_N)^T. \end{aligned} \tag{A.16}$$

Let components of Q_{NN} be denoted as $Q_{NN}^1, Q_{NN}^2, Q_{NN}^3, Q_{NN}^4$, we can obtain as follows;

$$\begin{aligned} Q_{NN}^1 &= \mathbf{X}_1 + \mathbf{X}_2 + \mathbf{X}_3 + \mathbf{X}_4 \\ Q_{NN}^2 &= \mathbf{X}_1 - \mathbf{X}_2 + \mathbf{X}_3 - \mathbf{X}_4 \\ Q_{NN}^3 &= \mathbf{X}_1 + \mathbf{X}_2 - \mathbf{X}_3 - \mathbf{X}_4 \\ Q_{NN}^4 &= \mathbf{X}_1 - \mathbf{X}_2 - \mathbf{X}_3 + \mathbf{X}_4 \end{aligned} \tag{A.17}$$

where $\mathbf{X}_i$ is a 2-D SWHT of sub-blocks defined as $T_{N/2}\mathbf{x}_i(T_{N/2})^T$. From Equation (A.17), we can see the relationship between Q_{NN} and $\mathbf{X_i}$ is 2×2 2-D SWHT as follows;

$$Q_{NN} = T_2 \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 \\ \mathbf{X}_3 & \mathbf{X}_4 \end{bmatrix} T_2^T \tag{A.18}$$

From Equation (A.18) and Equation (A.13), we obtain a compact form of Equation (A.16) using the property of permutation matrix ($(P_N^{4,m})^T = (P_N^{4,m})^{-1}$) and Kronecker product ($(a \otimes b)^{-1} = a^{-1} \otimes b^{-1}$, $(a \otimes b)(c \otimes d) = (ac \otimes bd)$) as follows;

$$\begin{aligned} X_{NN} &= P_N Q_{NN} (P_N)^T \\ &= P_N Q_{NN} (P_N)^{-1} \\ &= (I_{N/2^2} \otimes S_2) \overline{Q_{NN}} (I_{N/2^2}^{-1} \otimes S_2^{-1}) \\ &= (I_{N/2^2} \otimes S_2) \overline{Q_{NN}} (I_{N/2^2} \otimes S_2) \\ &= (I_{N/2^3} \otimes S_4^*) \overline{Q_{NN}} \end{aligned} \tag{A.19}$$

where $\overline{Q_{NN}}$ is reordered 2×2 SWHT of sub-blocks' coefficients defined as

$$\begin{aligned} Q_{NN} = \{&\overline{X}_1(0), \overline{X}_2(0), \overline{X}_3(0), \overline{X}_4(0), \ldots, \\ &\overline{X}_1(\frac{N}{2}-1), \overline{X}_2(\frac{N}{2}-1), \overline{X}_3(\frac{N}{2}-1), \overline{X}_4(\frac{N}{2}-1)\} \end{aligned} \tag{A.20}$$

and S_4^* is a reordering matrix denoted as

$$S_4^* = \begin{bmatrix} I_2 & \overline{0} & \overline{0} & \overline{0} \\ \overline{0} & B_2 & \overline{0} & \overline{0} \\ \overline{0} & \overline{0} & \overline{I_2} & \overline{0} \\ \overline{0} & \overline{0} & \overline{0} & B_2\overline{I_2} \end{bmatrix} \tag{A.21}$$

and B_2 is a vertical transition matrix satisfying the condition below.

$$\begin{bmatrix} x(1) & x(0) \\ x(3) & x(2) \end{bmatrix} = B_2 \begin{bmatrix} x(0) & x(1) \\ x(2) & x(3) \end{bmatrix} \tag{A.22}$$

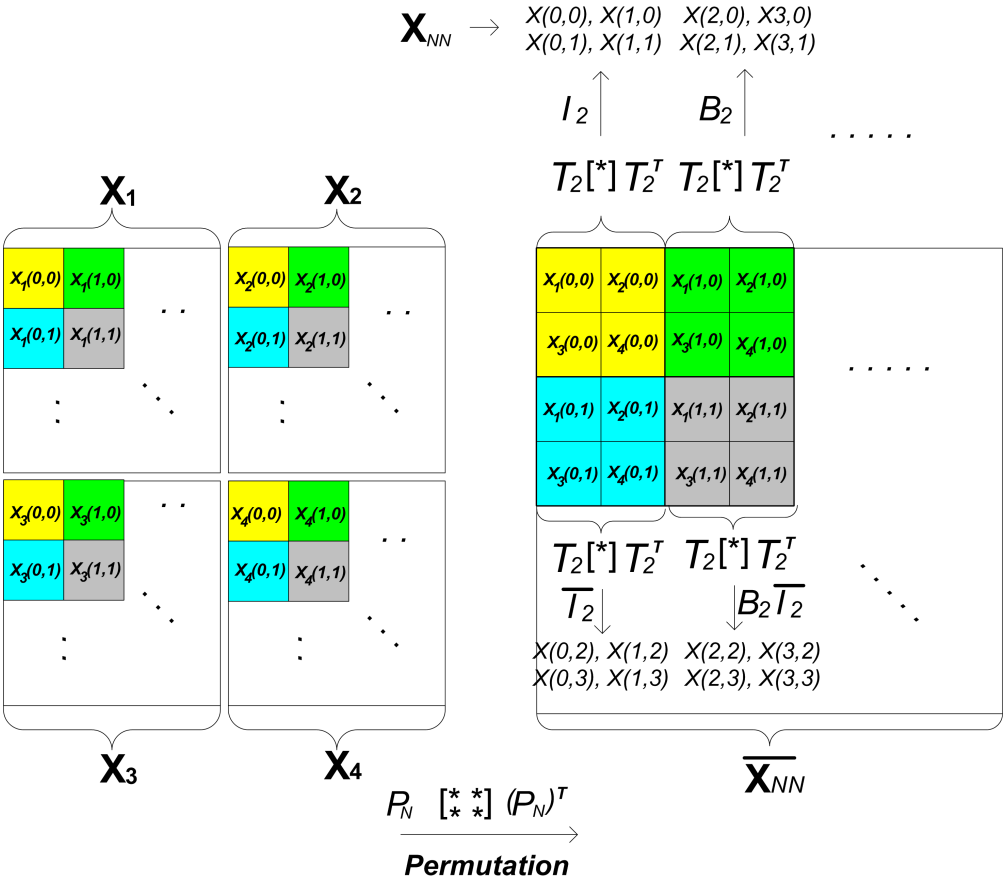


Figure A.2: Graphical representation of relationship between $N \times N$-point SWHT ($\mathbf{X}$) and $N/2 \times N/2$-point sub-blocks' SWHT

Figure A.2 shows the graphical procedure of two dimensional SWHT coefficients between a block and its sub-blocks described in Equation (A.19). We reorder coefficients of sub-blocks' SWHT so that coefficients located in the first quadrant of the Cartesian Coordinate System can be obtained only multiplying by the identity matrix. In a similar fashion, the coefficients of other quadrants are calculated by multiplying the reflection matrix ($\overline{I_2}$), horizontal transition matrix (B_2) and its multiplication ($B_2\overline{I_2}$) respectively.

To provide a comprehensive illustration of the generalized linear relationship between SWHT coefficients of blocks, Figure A.3 presents a full coefficients composition example between a 4×4 block and its four 2×2 sub-blocks. The summarized composition procedures are as follows and decomposition is also possible by inverse procedures.

1. Taking SWHT coefficients of separated four 2×2 blocks independently (Figure A.3(b)).
2. Reordering coefficients according to its position as shown in Figure A.3(c).
3. Separating 4×4 reordered block into four 2×2 blocks, whose positions are marked as $S_{i,j}$ (see Figure A.3(c)).
4. Calculating 4×4 block coefficients (see Figure A.3(d)) using Equation (A.19)

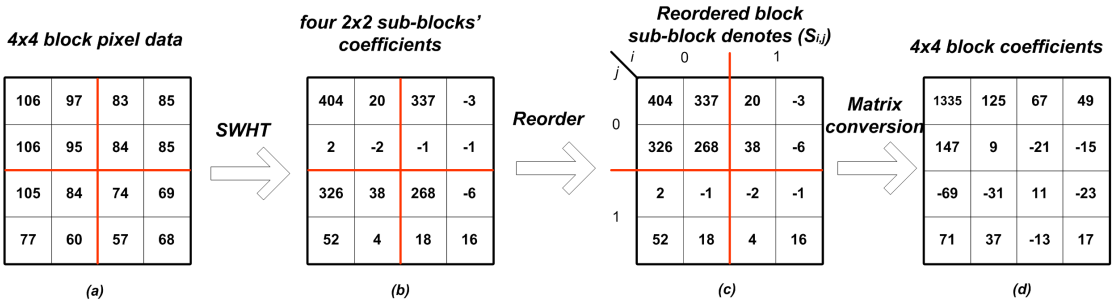


Figure A.3: Comprehensive illustration of SWHT coefficients relationship between four blocks of 2×2 and one block of 4×4 pixels

Bibliography

[1] *Available : http://ffmpeg.org/.*

[2] *Intel IPP Reference Manual : Volume2: Image and Video Processing.*

[3] *x264 Available:http://developers.videolan.org/x264.html.*

[4] E. Akyol, D. Mukherjee, and Yuxin Liu. Complexity control for real-time video coding. In *Proc. IEEE International Conference on Image Processing ICIP 2007*, volume 1, pages I–77–I–80, September 2007.

[5] P. Anandan. *Measurement Visual Motion from Image Sequences.* PhD thesis, University of Massachusetts, 1987.

[6] P.E. Anuta. Spatial registration of multispectral and multitemporal digital imagery using fast fourier transform techniques. *IEEE Trans. Geosci. Electron*, 8(4):353–368, 1970.

[7] H. F. Ates and Y. Altunbasak. Rate-distortion and complexity optimized motion estimation for h.264 video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(2):159–171, Feb 2008.

[8] A. Averbuch and Y. Keller. Fast motion estimation using bidirectional gradient methods. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '02)*, volume 4, pages IV–3616–IV–3619, May 13–17, 2002.

[9] Gisle Bjontegaard. Calculation of average psnr differences between rd-curves, April 2001.

[10] V. Bruni, D. De Canditiis, and D. Vitulano. Phase information and space filling curves in noisy motion estimation. *IEEE Trans. Image Process*, 18(7):1660–1664, July 2009.

[11] J. Cabrera, J. I. Ronda, A. Ortega, and N. Garcia. Stochastic rate-control of interframe video coders for VBR channels. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, volume 3, pages 813–6, September 2003.

[12] J.F. Canny. A computational approach to edge detection. *IEEE Trans Patten Analysis and Machine Intelligence*, 8:679–698, 1986.

[13] J. Chakareski, J. Apostolopoulos, and B. Girod. Low-complexity rate-distortion optimized video streaming. In *Image Processing, 2004. ICIP '04. 2004 International Conference on*, volume 3, pages 2055–2058, October 2004.

[14] M. H. Chan, Y. B. Yu, and A. G. Constantinides. Variable size block matching motion compensation with applicationsto video coding. In *Communications, Speech and Vision, IEE Proceedings I*, volume 137, pages 205–212, August 1990.

[15] Yui-Lam Chan and Wan-Chi Siu. An efficient search strategy for block motion estimation using image features. *IEEE Transactions on Image Processing*, 10(8):1223–1238, August 2001.

[16] Yi-Chih Chao, Kuan-Hung Lin, Bin-Da Liu, and Jar-Ferr Yang. An approximate square criterion for H.264/AVC intra mode decision. In *Multimedia and Expo, 2008 IEEE International Conference on*, pages 333–336, Hannover„ June/April 2008.

[17] Ching-Yeh Chen, Yu-Wen Huang, Chia-Lin Lee, and Liang-Gee Chen. One-pass computation-aware motion estimation with adaptive search strategy. *IEEE Trans. Multimedia*, 8(4):698–706, Aug 2006.

[18] Lien-Fei Chen, Shin-Ping Yang, and Yeong-Kang Lai. Model-based early termination scheme for h.264/avc inter prediction. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2009*, pages 597–600, April 19–24, 2009.

[19] Tihao Chiang and Ya-Qin Zhang. A new rate control scheme using Quadratic Rate Distortion Model. *IEEE Transactions on Circuits and Systems for Video Technology*, 7(1):246–250, February 1997.

[20] Inchoon Choi, Jeyun Lee, and Byeungwoo Jeon. Fast coding mode selection with rate-distortion optimization for mpeg-4 part-10 avc/h.264. *IEEE Transactions on Circuits and Systems for Video Technology*, 16(12):1557–1561, December 2006.

[21] Ciaran O Connaire, Philip Kelly, Chanyul Kim, and Noel O'Connor. Automatic camera selection for activity monitoring in a multi-camera system for tennis. In *ICDSC 2009 - 3rd ACM/IEEE International Conference on Distributed Smart Cameras*, 2009.

[22] R. Costantini, J. Bracamonte, G. Ramponi, J-L. Nagel, M. Ansorge, and F. Pell andini. A low-complexity video coder based on the discrete walsh hadamard transform. In *EUPSICO 2000 : European signal processing conference*, pages 1217–1220, 2000.

[23] Cheng Du and Yun He. Early detection of all zero chroma blocks in H.263. In *Proc. 5th International Conference on Signal Processing WCCC-ICSP 2000*, volume 2, pages 1110–1114, August 21–25, 2000.

[24] F. Dufaux and F. Moscheni. Motion estimation techniques for digital tv: a review and a new contribution. *Proc. IEEE*, 83(6):858–876, June 1995.

[25] N. Eiamjumrus and S. Aramvith. New rate control Scheme based on Cauchy Rate-Distortion Optimization Model for H.264 Video Coding. In *Intelligent Signal Processing and Communications, 2006. ISPACS '06. International Symposium on*, pages 143–146, Yonago, December 2006.

[26] F.Essannouni, Y.Hadi, Oulad Haj, and A.Salam. A new optimal frequency motion estimation algorithm. In *ISCCSP*, 2006.

[27] Y. A. Geadah and M. J. G. Corinthios. Natural, dyadic, and sequency order algorithms and processors for the walsh-hadamard transform. *IEEE Transactions on Computers*, 26(5):435–442, May 1977.

[28] H. Gharavi and M. Mills. *Block matching motion estimation algorithms-new results*. IEEE Transactions on Circuits and Systems, May 1989.

[29] Zhenghui Gu, Shoulie Xie, and S. Rahardja. Unified complex hadamard transform sequences for multi-carrier CDMA systems. In *Vehicular Technology Conference, 2004. VTC 2004-Spring. 2004 IEEE 59th*, volume 3, pages 1514–1517, May 2004.

[30] M. N. Gulamhusein. Simple matrix-theory proof of the discrete dyadic convolution theorem. *Electronics Letters*, 9:238–239, May 1973.

[31] Zhihai He, Yong Kwan Kim, and S. K. Mitra. Low-delay rate control for DCT video coding via p-domain source modeling. *IEEE Transactions on Circuits and Systems for Video Technology*, (8):928–940, August.

[32] Zhihai He, Yongfang Liang, Lulin Chen, I. Ahmad, and Dapeng Wu. Power-rate-distortion analysis for wireless video communication under energy constraints. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(5):645–658, May 2005.

[33] Y. Hel-Or and H. Hel-Or. Real-time pattern matching using projection kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(9):1430–1445, September 2005.

[34] Yu Hu, Qing Li, S. Ma, and C.-C. Jay Kuo. Joint rate-distortion-complexity optimization for h.264 motion search. In *Proc. IEEE International Conference on Multimedia and Expo*, pages 1949–1952, July 9–12, 2006.

[35] Yu-Wen Huang, Ching-Yeh Chen, Chen-Han Tsai, Chun-Fu Shen, and Liang-Gee Chen. Survey on block matching motion estimation algorithms and architectures with new results. *J. VLSI Signal Process. Syst.*, 42(3):297–320, 2006.

[36] En hui Yang and Xiang Yu. Rate distortion optimization of h.264 with main profile compatibility. In *Proc. IEEE International Symposium on Information Theory*, pages 282–286, July 9–14, 2006.

[37] O Hunt and R. Mukundan. A Comparison of Discrete Orthogonal Basis Functions for Image Compression. In *Image and Vision Computing New Zealand (IVCNZ-2004)*, pages 53–58, 2004.

[38] S. Ishwar, P. K. Meher, and M. N. S. Swamy. Discrete tchebichef transform-a fast 4x4 algorithm and its application in image/video compression. In

Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on, pages 260–263, Seattle, WA, May 2008.

[39] Y. V. Ivanov and C. J. Bleakley. Skip prediction and early termination for fast mode decision in h.264/avc. In *Proc. International Conference on Digital Telecommunications, ICDT '06*, page 7, August 29–31, 2006.

[40] N.S. Jayant and P.Noll. *Digital Coding of Waveforms.* Prentice-Hall, NJ, 1984.

[41] C. S. Kannangara, I. E. Richardson, and A. J. Miller. Computational complexity management of a real-time H.264/AVC encoder. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(9):1191–1200, September 2008.

[42] Chaminda Sampath Kannangara. *Complexity Management of H.264/AVC Video Compression.* PhD thesis, The Robert Gordon University, 2006.

[43] K Karhunen. Uber lineare methoden in der wahrscheinlich-keitsrechnung. *Ann. Acad. Sci. Fennicea*, A137, 1947. Translated by Selin, I. in "On Linear Methods in Probability Theory," Doc. T-131,The RAND Corp, Santa Monica,CA,1960.

[44] Y. Keller and A. Averbuch. Fast motion estimation using bidirectional gradient methods. *IEEE Trans. Image Process*, 13(8):1042–1054, August 2004.

[45] S. L. Kilthau, M. S. Drew, and T. Moller. Full search content independent block matching based on the fast fourier transform. In *Proc. International Conference on Image Processing 2002*, volume 1, pages I–669–I–672, September 22–25, 2002.

[46] Chanyul Kim and Noel O'Connor. Using the discrete hadamard transform to detect moving objects in surveillance video. In *VISAPP 2009 - International Conference on Computer Vision Theory and Applications*, 2009.

[47] Sung Deuk Kim, Jaeyoun Yi, Hyun Mun Kim, and Jong Beom Ra. A deblocking filter with two separate modes in block-based video coding.

IEEE Transactions on Circuits and Systems for Video Technology, 9(1):156–160, February 1999.

[48] Donald E. Knuth. *Art of Computer Programming, Volume 3: Sorting and Searching (2nd Edition)*. Addison-Wesley Professional, April 1974.

[49] Ut-Va Koc and K.J. Ray Liu. Motion compensation on dct domain. *EURASIP Journal on Applied Signal Processing*, 2001:147–162, 2001.

[50] T Koga, K. Ilinuma, A. Hirano, Y.Iijima, and T.Ishiguro. Motion compensated interframe coding for video conferencing. In *in Proc Nat. telecommum. Conf*, pages G5.3.1–G5.3.5, New Orleans, 1981.

[51] Lauri Koskinen, Ari Paasio, and Kari Halonen. Cnn-type algorithms for H.264 variable block-size partitioning. *Image Commun.*, 22(9):797–808, 2007.

[52] Peter Kuhn. *Algorithms,complexity analysis and vlsi architectures for MPEG-4 motion estimation*. Kluwer Academic Publishers, Boston, 1999.

[53] Dmitriy Kulikov and Alexander Parshin. Mpeg-4 avc/h.264 video codecs comparision. Technical report, MSU, May 2009.

[54] S. Kumar, M. Biswas, and T. Q. Nguyen. Efficient phase correlation motion estimation using approximate normalization. In *Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers*, volume 2, pages 1727–1730, November 7–10, 2004.

[55] Tien-Ying Kuo and Hsin-Ju Lu. Efficient h.264 encoding based on skip mode early termination. *LNCS, Advance in Image and Video Technology*, 4319:761–770, Dec 2006.

[56] M. Li, M. Biswas, S. Kumar, and Truong Nguyen. Dct-based phase correlation motion estimation. In *Proc. International Conference on Image Processing ICIP '04*, volume 1, pages 445–448, October 24–27, 2004.

[57] Reoxiang Li, Bing Zeng, and M. L. Liou. A new three-step search algorithm for block motion estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, 4(4):438–442, August 1994.

[58] Xiang Li, N. Oertel, A. Hutter, and A. Kaup. Advanced lagrange multiplier selection for hybrid video coding. In *Proc. IEEE International Conference on Multimedia and Expo*, pages 364–367, July 2–5, 2007.

[59] B. Liu and A. Zaccarin. New fast algorithms for the estimation of block motion vectors. *Circuits and Systems for Video Technology, IEEE Transactions on*, 3(2):148–157, 1993.

[60] Lurng-Kuo Liu and E. Feig. A block-based gradient descent search algorithm for block motion estimation in video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 6(4):419–422, August 1996.

[61] Qin LIU, Yiqing HUANG, Satoshi GOTO, and Takeshi IKENAGA. Edge block detection and motion vector information based fast vbsme algorithm. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences Advance Access*, E91(A), Aug 2008.

[62] L.Lgai, M.C.Man, and C.W.Kuen. Fast block matching algorithm in walsh hadamard domain. In *Asian Conference on Computer Vision(ACCV)*, pages 712–721, Hyderabad, India, Jan 2006.

[63] H. Lohscheller. A subjectively adapted image communication system. *IEEE Transactions on Communications*, 32(12):1316–1322, December 1984.

[64] F. J. P. Lopes and M. Ghanbari. Analysis of spatial transform motion estimation with overlapped compensation and fractional-pixel accuracy. *IEE Proceedings -Vision, Image and Signal Processing*, 146(6):339–344, December 1999.

[65] Meng-Ting Lu, Jason J. Yao, and Homer H. Chen. A Complexity-Aware Video Adaptation Mechanism for Live Streaming Systems. *EURASIP Journal on Advances in Signal Processing*, 2007:1–10, 2007.

[66] Xiaoan Lu, A. M. Tourapis, Peng Yin, and J. Boyce. Fast mode decision and motion estimation for h.264 with a focus on MPEG-2/h.264 transcoding. In *Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on*, pages 1246–1249, May 2005.

[67] Xiaoan Lu, Yao Wang, and E. Erkip. Power efficient h.263 video transmission over wireless channels. In *Proc. International Conference on Image Processing 2002*, volume 1, pages I–533–I–536, September 22–25, 2002.

[68] Cooltheart M. The persistences of vision. *Philos Trans R Soc Lond B Biol Sci*, 8:57–69, Jul 1980.

[69] H. S. Malvar, A. Hallapuro, M. Karczewicz, and L. Kerofsky. Low-complexity transform and quantization in h.264/AVC. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):598–603, July 2003.

[70] H. Y. L. Mar and C. L. Sheng. Fast hadamard transform using the h diagram. *IEEE Transactions on Computers*, 22(10):957–960, October 1973.

[71] P. Marti-Puig. A family of fast walsh hadamard algorithms with identical sparse matrix factorization. *IEEE Signal Processing Letters*, 13(11):672–675, November 2006.

[72] Vinod Menezes, S. K. Nandy, and Biswadip Mitra. Signal compression through spatial frequency-based motion estimation. *Integr. VLSI J.*, 22(1-2):115–135, 1997.

[73] Loren Merritt. *X264: A HIGH PERFORMANCE H.264/AVC ENCODER.*

[74] Joan L. Mitchell, William B. Pennebaker, Chad E. Rogg, and Didier J. LeGall. *MPEG video compression standard.* Kluwer Academic Publishers, 1996.

[75] M.J.Corinthios. A time-series analyzer. *Computer Processing in Communication*, pages 47–69, Apr 1969.

[76] Yong Ho Moon, Gyu Yeong Kim, and Jae Ho Kim. An improved early detection algorithm for all-zero blocks in H.264 video encoding. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(8):1053–1057, August 2005.

[77] Y. Moshe and H. Hel-Or. A fast block motion estimation algorithm using gray code kernels. In *Proc. IEEE International Symposium on Signal Processing and Information Technology*, pages 185–190, August 2006.

[78] Kathy T. Mullen. The contrast sensitivity of human color vision to red-green and blue-yellow chromatic gratings. *J.Physiol*, 359:381–400, 1985.

[79] H. G. Musmann, P. Pirsch, and H.-J. Grallert. Advances in picture coding. *Proc. IEEE*, 73(4):523–548, April 1985.

[80] M.Viitanen, P.Kolinummi, T. Hamalainen, and J.Saarinen. Scalable dsp implementation of dct-based motion estimation algortithm. In *Eurasip*, 2000.

[81] K. Nakagaki and R. Mukundan. A fast 4x4 forward discrete tchebichef transform algorithm. *IEEE Signal Processing Letters*, 14(10):684–687, October 2007.

[82] Laplacian Source Nasir. Simulation of the rate-distortion behaviour of a memoryless, 2002.

[83] A.N. Netravali and J.D. Robbins. Motion compensated television coding. *B.S.T.J*, 58(3):1735–1745, Mar 1979.

[84] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, and T. Wedi. Video coding with h.264/AVC: tools, performance, and complexity. *IEEE Circuits and Systems Magazine*, 4(1):7–28, / 2004.

[85] Wei-Hau Pan, Shou-Der Wei, , and Shang-Hong Lai. Efficient ncc-based image matching in walsh-hadamard domain. *Lecture Notes in Computer Science*, pages 468–480, Oct 2008.

[86] P.Nillius and J.O. Eklundh. Fast block matching with normalized cross-correlation using walsh transforms. *Computational Vision and Active Perception Laboratory (CVAP)*, 2002.

[87] Lai-Man Po and Wing-Chung Ma. A novel four-step search algorithm for fast block motion estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, 6(3):313–317, June 1996.

[88] W. K. Pratt, J. Kane, and H. C. Andrews. Hadamard transform image coding. *Proceedings of the IEEE*, 57(1):58–68, January 1969.

[89] A. Puri, H. M. Hang, and D. Schilling. An efficient block-matching algorithm for motion-compensated coding. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '87.*, volume 12, pages 1063–1066, April 1987.

[90] F Rocca. Television bandwidth compression utilizing frame-to-frame correlation and movement compensation. In *Symposium on Picture Bandwidth Compression*, Gordon and Breach, NJ, 1972.

[91] R. Kh. Sadykhov, V. A. Samokhval, and L. P. Podenok. Face recognition algorithm on the basis of truncated walsh-hadamard transform and synthetic discriminant functions. In *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, pages 219–222, May 2004.

[92] A. Saha, Kallol Mallick, J. Mukherjee, and Shamik Sural. SKIP prediction for fast rate distortion optimization in h.264. *IEEE Transactions on Consumer Electronics*, 53(3):1153–1160, August 2007.

[93] H. Schwarz, D. Marpe, and T. Wiegand. Overview of the scalable video coding extension of the h.264/avc standard. *Circuits and Systems for Video Technology, IEEE Transactions on*, 17(9):1103–1120, September 2007.

[94] A.J. Seyler. The coding of visual signals to reduce channel-capacity requirements. *Proc. IEE, Part C*, 109:676–684, 1962.

[95] Miao Sima, Yuanhua Zhou, and Wei Zhang. An efficient architecture for adaptive deblocking filter of h.264/avc video coding. *IEEE Trans. Consum. Electron.*, 50(1):292–296, February 2004.

[96] Stephen Smoot and Lawrence A. Rowe. Laplacian model for ac dct terms in image and video coding. In *Proceedings of the 13th International Workshop on Network and Operating Systems Support for Digital Audio and Video Table of Contents, MontereyCA*, pages 60–69, 1996.

[97] SMPTE. *VC-1 Compressed Video Bitstream Format and Decoding Process*. SMPTE421M.

[98] L. A. Sousa. General method for eliminating redundant computations in video coding. *Electronics Letters*, 36(4):306–307, February 17, 2000.

[99] D. M Sow and A. Eleftheriadis. Complexity distortion theory. In *Proc. IEEE International Symposium on Information Theory 1997*, page 188, June 29–July 4, 1997.

[100] D. M. Sow and A. Eleftheriadis. Complexity distortion theory. *IEEE Transactions on Information Theory*, 49(3):604–608, March 2003.

[101] S.Pei, M. Du, and H.Feng. A modified method for detecting all-zero dct coefficients blocks before dct and quantization. *Journal of Infomation & Computational Science*, 1:263–268, 2004.

[102] R. Srinivasan and K. Rao. Predictive coding based on efficient motion estimation. *IEEE Transactions on Communications*, 33(8):888–896, August 1985.

[103] Gary J. Sullivan and Thomas Wiegand. Rate-Distortion Optimization for video compression. *IEEE Signal Processing Magazine*, pages 74–90, Nov 1998.

[104] G.J. Sullivan, P. Topiwala, and A. Luthra. The h.264/avc advanced video coding standard: Overview and introduction to the fidelity range extensions. Technical report, SPIE Conference on Applications of Digital Image Processing XXVII, Special Session on Advances in the New Emerging Standard: H.264/AVC, 2004.

[105] Yu Ting Sun and Yinyi Lin. SATD-based intramode decision for h.264/AVC video coding. In *Multimedia and Expo, 2008 IEEE International Conference on*, pages 61–64, Hannover,, June/April 2008.

[106] Pol-Lin Tai, Shih-Yu Huang, Chii-Tung Liu, and Jia-Shung Wang. Computation-aware scheme for software-based block motion estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(9):901–913, September 2003.

[107] Jo Yew Tham, S. Ranganath, M. Ranganath, and A. A. Kassim. A novel unrestricted center-biased diamond search algorithm for block motion estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(4):369–377, August 1998.

[108] P.H.S. Torr and A. Zisserman. Feature based methods for structure and motion estimation. *Lecture Notes in Computer Science, Vision Algorithms*, pages 278–295, 2000.

[109] D. Tzovaras, M.G. Strintzis, and H.Sahinolou. Evaluation of multiresolution block matching techinques for motion and disparity estimation. In *Signal Process. Image Ccommum.*, pages 56–67, 1994.

[110] M. van der Schaar and Y. Andreopoulos. Rate-distortion-complexity modeling for network and receiver aware adaptation. *IEEE Transactions on Multimedia*, 7(3):471–479, June 2005.

[111] R. Vanam, E. A. Riskin, S. S. Hemami, and R. E. Ladner. Distortion-complexity optimization of the h.264/MPEG-4 AVC encoder using the GBFOS algorithm. In *Data Compression Conference, 2007. DCC '07*, pages 303–312, Snowbird, UT, March 2007.

[112] Rahul Vanam, Eve A. Riskin, and Richard E. Ladner. H.264/MPEG-4 AVC encoder parameter selection algorithms for complexity distortion tradeoff. In *Data Compression Conference, 2009. DCC '09.*, pages 372–381, Snowbird, Utah, USA, March 2009.

[113] VanNes, Floris L., Bouman, and Maarten A. Spatial modulation transfer in the human eye. *Journal of the Optical Society of America*, 57:401–406, 1967.

[114] Floris L. VanNes, Bouman, and Maarten A. Spatial modulation transfer in the human eye. *Journal of the Optical Society of America*, 57:401–406, 1967.

[115] Shuai Wan, Fuzheng Yang, Mingyi He, and Ebroul Izquierdo. Rate distortion optimised motion estimation based on a general framework. In *Visual Information Engineering, 2008. VIE 2008. 5th International Conference on*, pages 83–87, Xian China, July/August 2008.

[116] H. M. Wang, C. H. Tseng, and J. F. Yang. Computation reduction for intra 4x4 mode decision with SATD criterion in h.264/AVC. *IET Signal Processing*, 1(3):121–127, September 2007.

[117] Hanli Wang, S. Kwong, and C. W. Kok. Efficient prediction algorithm of integer DCT coefficients for h.264/AVC optimization. *IEEE Transactions*

on Circuits and Systems for Video Technology, 16(4):547–552, April 2006.

[118] Hanli Wang, S. Kwong, and Chi-Wah Kok. Analytical model of zero quantized dct coefficients for video encoder optimization. In *Proc. IEEE International Conference on Multimedia and Expo*, pages 801–804, July 9–12, 2006.

[119] Hanli Wang and Sam Kwong. Hybrid model to detect zero quantized dct coefficients in h.264. *IEEE Trans. Multimedia*, 9(4):728–735, June 2007.

[120] Hanli Wang and Sam Kwong. Prediction of zero quantized dct coefficients in h.264/avc using hadamard transformed information. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(4):510–515, April 2008.

[121] Miaohui Wang and Bo Yan. Lagrangian multiplier based joint three-layer rate control for h.264/avc. *IEEE Signal Processing Letters*, 16(8):679–682, August 2009.

[122] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4):600–612, April 2004.

[123] 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*, 13(7):560–576, July 2003.

[124] Zhou Xuan, Yu Zhenghua, and Yu Songyu. Method for detecting all-zero DCT coefficients ahead of discretecosine transformation and quantisation. *Electronics Letters*, 34(19):1839–1840, September 1998.

[125] X.Yi, J.Zhang, N.Ling, and W.Shang. Improved and simplified fast motion estimation for jm. Technical report, Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T, Poznan, Poland, Jul 2005.

[126] En-Hui Yang and Xiang Yu. Rate distortion optimization for h.264 interframe coding: A general framework and algorithms. *IEEE Transactions on Image Processing*, 16(7):1774–1784, Jul 2007.

[127] Jar-Ferr Yang, Shih-Cheng Chang, and Chin-Yun Chen. Computation reduction for motion search in low rate video coders. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(10):948–951, October 2002.

[128] Libo Yang, K. Yu, Jiang Li, and Shipeng Li. An effective variable block-size early termination algorithm for H.264 video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(6):784–788, June 2005.

[129] Xiaoquan Yi and Nam Ling. Scalable complexity-distortion model for fast motion estimation. In *Proc. SPIE*, 2006.

[130] Lu Yu and Feng Yi. Low-complexity in-loop deblocking filter. Technical report, JVT VEDG, 2005.

[131] N. H. C. Yung and W. H. Mok. A novel and fast feature based motion estimation algorithm through extraction of the background and moving objects. In *Circuits and Systems, 1998. ISCAS '98. Proceedings of the 1998 IEEE International Symposium on*, volume 4, pages 138–141, Monterey, CA, May/June 1998.

[132] Z.Chen, P.Zhou, and Y.He. Fast integer and fractional pel motion estimation for jvt, 2002.

[133] Shan Zhu and Kai-Kuang Ma. A new diamond search algorithm for fast block matching motion estimation. In *Information, Communications and Signal Processing, 1997. ICICS., Proceedings of 1997 International Conference on*, volume 1, pages 292–296, September 1997.

[134] Z.Wang and A.C.Bovik. Mean squared error: love it or leave it? - a new look at signal fidelity measures. *IEEE Signal Processing Magazine*, 26(1):98–117, Jan 2009.