

# Computational Complexity Management of H.264/AVC Video Coding Standard

*Serdar Burak Solak*



Department of Electrical & Computer Engineering  
McGill University  
Montreal, Canada

October 2010

---

A thesis submitted to McGill University in partial fulfillment of the requirements for the degree of Masters of Engineering.

© 2010 Serdar Burak Solak

## Abstract

The new H.264/AVC video coding standard achieves significantly improved compression efficiency compared to previous standards by adopting highly advanced and flexible encoding techniques at the expense of increased complexity. However, the high computational complexity of H.264/AVC is a big concern primarily for low-power devices with limited processing capabilities. This thesis presents new techniques to reduce and/or control the computational complexity of an H.264/AVC encoder.

A new prediction method is developed to estimate the Lagrangian rate-distortion cost of a macroblock. The prediction method is used in the design of two complexity reduction algorithms for H.264/AVC. The first algorithm uses the predicted rate-distortion costs to identify the SKIP coded macroblocks prior to any INTRA or INTER mode trial. Simulation results show that the algorithm achieves significant complexity savings with negligible loss in rate-distortion performance. Similarly, the second algorithm seeks to further reduce the encoder complexity by using the predicted costs to identify not only SKIP coded but also the INTRA and INTER coded macroblocks at earlier stages. Results indicate greater reductions in the encoder complexity at the expense of slightly larger loss in rate-distortion performance.

A complexity scalable encoding framework is proposed for controlling the encoder complexity at a macroblock level using a single parameter. The framework uses a special macroblock grouping technique called the “wave-front macroblock scheduling”. The computational resources are allocated to the macroblocks within a wave-front. The resource allocation is further developed by adopting the Lagrangian rate-distortion cost prediction into the framework. Results demonstrate significant improvements in the rate-distortion performance of the encoder operating at limited complexity. Finally, the complexity reduction algorithms are installed into the complexity scalable encoding framework. Simulations show that these algorithms equip the complexity scalable encoder with additional complexity control.

These novel algorithms are designed with the target of enabling the H.264/AVC implementations in computationally constrained environments such as the hand-held devices with limited processing capabilities and limited battery life.

## Sommaire

La norme de codage vidéo H.264/AVC permet une efficacité de compression grandement supérieure à celle des normes précédentes grâce à des techniques de codage avancées d'une grande flexibilité. Ceci dit, le prix de cette performance améliorée est l'augmentation de la complexité du calcul requise, ce qui est un obstacle majeur pour les appareils dont la puissance et la capacité de calcul sont limitées. Ce mémoire présente de nouvelles techniques pour réduire et contrôler la complexité du calcul requise par un codeur H.264/AVC.

Une nouvelle méthode de prédiction est développée pour estimer le coût débit-distorsion Lagrangien d'un macrobloc. Cette méthode est utilisée avec deux nouveaux algorithmes de réduction de la complexité pour un codeur H.264/AVC. Le premier algorithme utilise les coûts prédits du taux de distorsion pour identifier les macroblocs codés de type SKIP avant les essais des modes INTRA ou INTER. Des simulations démontrent que cet algorithme entraîne une réduction significative de la complexité du calcul avec une diminution négligeable de la performance débit-distorsion. Le deuxième algorithme utilise la méthode de prédiction des coûts débit-distorsion pour réduire la complexité du codeur en identifiant les macroblocs codés de type INTRA et INTER plus tôt lors du processus de codage. Les résultats indiquent que des réductions encore plus grandes de la complexité peuvent être obtenues au prix d'une dégradation accrue de la performance débit-distorsion.

Un dispositif de contrôle évolutif est proposé pour contrôler la complexité au niveau du macrobloc à l'aide d'un unique paramètre. Le dispositif utilise une technique de regroupement gérant l'allocation des ressources de calcul aux macroblocs et intègre la méthode de prédiction du coût débit-distorsion Lagrangien. Les résultats démontrent une amélioration significative de la performance du taux de distorsion tout en limitant la complexité. Finalement, les algorithmes de réduction de la complexité sont ajoutés au dispositif de contrôle, ce qui permet un meilleur contrôle de la complexité utilisée lors du codage.

Ces nouveaux algorithmes sont conçus pour permettre l'implantation de la norme H.264/AVC dans des environnements comme les appareils sans-fil utilisant des batteries à autonomie limitée ainsi qu'une capacité de calcul limitée.

## Acknowledgments

I take this opportunity to express my sincere thanks to all who helped me to complete this work.

First and foremost, I am beyond grateful to my supervisor Prof. Fabrice Labeau for his constant guidance, encouragement, patience, support and most importantly for giving me the opportunity to carry out this work. I am thankful for the privilege of working with him.

Thanks to all my friends and colleagues in the TSP labs for their support and assistance. I would like to extend my deepest gratitude to Sunday Nyamweno and Ramdas Satyan.

Finally, I would like to give my special thanks to my parents for their continuous love, support and encouragement.

---

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Video Compression Basics . . . . .	2
1.2	Problem of Computational Complexity in Video Compression . . . . .	3
1.3	Thesis Contribution . . . . .	4
1.4	Thesis Overview . . . . .	4
<b>2</b>	<b>Overview and Complexity Analysis of H.264/AVC</b>	<b>6</b>
2.1	Video Encoding Concepts . . . . .	7
2.2	H264/AVC Encoder . . . . .	10
2.2.1	Mode Decision . . . . .	11
2.2.2	Transform . . . . .	17
2.2.3	Quantization . . . . .	17
2.2.4	Entropy Coder . . . . .	18
2.2.5	Deblocking Filter . . . . .	19
2.3	Encoder Complexity Management Algorithms . . . . .	19
2.3.1	Low Complexity Motion Estimation Algorithms . . . . .	19
2.3.2	Low Complexity Mode Decision . . . . .	21
2.3.3	Complexity Scalable Algorithms . . . . .	23
2.4	Summary . . . . .	24
<b>3</b>	<b>Experimental Method</b>	<b>26</b>
3.1	Video Encoder Software . . . . .	26
3.2	Development Environment . . . . .	27
3.3	Test Platform . . . . .	27
3.4	Test Video Sequences . . . . .	28

---

3.5	Testing Methodology and Performance Evaluation . . . . .	29
3.5.1	Video Quality . . . . .	29
3.5.2	Bit-rate . . . . .	32
3.5.3	Computational Complexity . . . . .	32
3.6	Summary . . . . .	33
<b>4</b>	<b>Complexity Reduction Tools</b>	<b>34</b>
4.1	Introduction . . . . .	34
4.2	RD-cost Prediction . . . . .	37
4.3	Application of RD-cost Prediction on Early Skip Termination . . . . .	40
4.3.1	Experimental Results . . . . .	40
4.4	Application of RD-cost Prediction on Early Mode Termination . . . . .	44
4.4.1	Experimental Results . . . . .	46
4.5	Summary . . . . .	46
<b>5</b>	<b>The Complexity Scalable Encoding Framework</b>	<b>49</b>
5.1	Introduction . . . . .	49
5.2	The Complexity Scalable Encoding Framework . . . . .	51
5.2.1	Experimental Results . . . . .	55
5.3	Improved Resource Allocation: Application of RD-cost Prediction in Scalable Encoding . . . . .	60
5.3.1	Experimental Results . . . . .	62
5.4	Complexity Reduction Extension: Application of EST & EMT in Scalable Encoding . . . . .	67
5.4.1	Experimental Results . . . . .	71
5.5	Summary . . . . .	77
<b>6</b>	<b>Conclusion</b>	<b>79</b>
6.1	Summary . . . . .	79
6.2	Future Work . . . . .	80
	<b>Bibliography</b>	<b>82</b>

# List of Figures

1.1	Illustration of spatial correlations with in a frame. . . . .	2
1.2	Illustration of temporal correlations between two consecutive frames. . . . .	3
2.1	Scopes of the H.264/AVC standard and this work. . . . .	6
2.2	The block diagram of a generic hybrid video encoder. . . . .	7
2.3	The block diagram of a typical H.264 Encoder. . . . .	9
2.4	An arbitrary frame divided into macroblocks each of size $16 \times 16$ pixels. . . . .	10
2.5	The supported INTRA_4 $\times$ 4 modes. . . . .	13
2.6	The supported INTRA_16 $\times$ 16 modes. . . . .	13
2.7	The supported INTER prediction modes. . . . .	15
2.8	The illustration of half and quarter pixel interpolation from integer pixel positions. . . . .	16
2.9	Transform of a block of residual pixels into a set of transform coefficients. . . . .	18
2.10	Quantization of a set of transform coefficients. . . . .	18
3.1	Snapshots of the 29 <sup>th</sup> Frame of the 6 sequences Akiyo (a), Football (b), Foreman (c), NBA (d), Silent (e) and Stefan (f). . . . .	30
3.2	The block diagram of a test scenario. . . . .	31
4.1	Average percentage distribution of computational time spent evaluating each mode. The numbers are rounded for the clarity of the graph. . . . .	35
4.2	Average percentage of number of macroblocks coded in each mode. The numbers are rounded for the clarity of the graph. . . . .	36
4.3	Optimal $\alpha_d$ and its second degree polynomial estimation. . . . .	38
4.4	Optimal $\alpha_r$ and its second degree polynomial estimation. . . . .	39

4.5	Histogram of the absolute macroblock RD cost prediction error percentages for 15 different sequences and 17 QPs. . . . .	39
4.6	Block diagram of the Early SKIP Termination algorithm . . . . .	41
4.7	RD curves of the proposed EST method . . . . .	42
4.8	RDO Mode decision representation by distance-between-the-curves method as QP varies for Football sequence . . . . .	42
4.9	Block diagram of the Early Mode Termination algorithm . . . . .	46
4.10	RD curves of the proposed EMT method . . . . .	47
5.1	MBs in wave-front #11 are highlighted. . . . .	52
5.2	The Block diagram of The Complexity Scalable Encoding of Frame $F_n$ . . .	54
5.3	Total encoding time in seconds vs. $\beta$ curves for the complexity scalable framework proposed by Tan, Lee, Tham and Rahardja [23]. . . . .	56
5.4	Total encoding time in seconds vs. $\beta$ curves for our proposed complexity scalable encoding framework. . . . .	56
5.5	Total encoding time in seconds vs. $\beta$ curves compared for our proposed complexity scalable encoding framework and for the complexity scalable framework proposed by Tan, Lee, Tham and Rahardja [23]. . . . .	57
5.6	RD curves for the proposed complexity scalable encoder for <i>Akiyo</i> and <i>Silent</i> , each curve is comprised of 11 $\beta$ values ranging from 0.0 to 1.0 by increments of 0.1. . . . .	58
5.7	RD curves for the proposed complexity scalable encoder for <i>Foreman</i> and <i>Football</i> , each curve is comprised of 11 $\beta$ values ranging from 0.0 to 1.0 by increments of 0.1. . . . .	59
5.8	Mode decision representation by distance-between-the-curves method as $\beta$ varies for Foreman with QP of 24. . . . .	59
5.9	Histogram of the macroblock RD cost values for Foreman sequence with a QP of 28. . . . .	60
5.10	The Block diagram of The Complexity Scalable Encoding of Frame $F_n$ with Improved Resource Allocation. . . . .	63
5.11	PSNR and Bit-rate gains with improved resource allocation over different $\beta$ values for Akiyo sequence . . . . .	64

5.12 PSNR and Bit-rate gains with improved resource allocation over different $\beta$ values for Silent sequence . . . . .	64
5.13 PSNR and Bit-rate gains with improved resource allocation over different $\beta$ values for Foreman sequence . . . . .	64
5.14 PSNR and Bit-rate gains with improved resource allocation over different $\beta$ values for Football sequence . . . . .	65
5.15 Total encoding time in seconds vs. $\beta$ curves for the complexity scalable encoder with improved resource allocation. . . . .	65
5.16 RD curves with improved resource allocation for <i>Football</i> , each curve is comprised of 10 $\beta$ values ranging from 0.1 to 1.0 by increments of 0.1. . . . .	66
5.17 Mode decision comparison of with the improved resource allocation (b) and without the improved resource allocation (a) by distance-between-the-curves method between over different $\beta$ values for Football sequence. . . . .	66
5.18 Mode decision comparison of with the improved resource allocation (b) and without the improved resource allocation (a) by distance-between-the-curves method between over different $\beta$ values for Foreman sequence. . . . .	67
5.19 The Block diagram of The Complexity Scalable Encoding of Frame $F_n$ with EST. . . . .	69
5.20 The Block diagram of The Complexity Scalable Encoding of Frame $F_n$ with EMT. . . . .	72
5.21 Total encoding time in seconds vs. $\beta$ curves for Foreman sequence with encoders: the complexity scalable encoding framework (CSEF), CSEF-EST and CSEF-EST. . . . .	74
5.22 Total encoding time in seconds vs. $\beta$ curves for Football sequence with encoders: the complexity scalable encoding framework (CSEF), CSEF-EST and CSEF-EST. . . . .	74
5.23 RD curves of Foreman sequence with varying $\beta$ points for full budget CSEF, CSEF-EST and CSEF-EMT encoders per Equation 5.1. . . . .	74
5.24 RD curves of Foreman sequence with varying $\beta$ points for reduced budget CSEF, CSEF-EST and CSEF-EMT encoders per Equation 5.2. . . . .	75
5.25 RD curves of Football sequence with varying $\beta$ points for full budget CSEF, CSEF-EST and CSEF-EMT encoders per Equation 5.1. . . . .	75

---

5.26	RD curves of Foreman sequence with varying $\beta$ points for reduced budget CSEF, CSEF-EST and CSEF-EMT encoders per Equation 5.2. . . . .	75
5.27	Mode decision comparison by distance-between-the-curves method of full budget CSEF, CSEF-EST and CSEF-EMT encoders per Equation 5.1 against varying $\beta$ points for Foreman sequence with a QP of 24. . . . .	76
5.28	Mode decision comparison by distance-between-the-curves method of reduced budget CSEF, CSEF-EST and CSEF-EMT encoders per Equation 5.2 against varying $\beta$ points for Foreman sequence with a QP of 24. . . . .	76
5.29	Mode decision comparison by distance-between-the-curves method of full budget CSEF, CSEF-EST and CSEF-EMT encoders per Equation 5.1 against varying $\beta$ points for Football sequence with a QP of 24. . . . .	76
5.30	Mode decision comparison by distance-between-the-curves method of reduced budget CSEF, CSEF-EST and CSEF-EMT encoders per Equation 5.2 against varying $\beta$ points for Football sequence with a QP of 24. . . . .	77

---

## List of Tables

4.1	Comparison of The Saved Encoding Time Percentages by The Proposed EST and by The EST Adopted in JM Reference Software. The Total Encoding Time for JSVM and JM encoders in RDO mode is also presented. . . . .	43
4.2	Comparison of The Relative Percentages of The Skipped MBs in EST and in RDO Mode Decision . . . . .	44
4.3	The Definition of The Mode Mapping Function <i>ModeToTest</i> . . . . .	45
4.4	Comparison of The Saved Encoding Time by The Proposed EST and by The Proposed EMT . . . . .	47
5.1	The Definition of The Mode Mapping Function <i>NextModeToTest</i> . . . . .	53

# List of Acronyms

AVC	Advanced Video Coding
CABAC	Context Adaptive Binary Arithmetic Coding
CAVLC	Context Adaptive Variable Length Coding
CIF	Common Intermediate Format
CSEF	Complexity Scalable Encoding Framework
EST	Early SKIP Termination
EMT	Early Mode Termination
EPZS	Enhanced Predictive Zonal Search
dB	Decibel
FSA	Full Search Algorithm
ITU-T	The Standardization Sector of The International Telecommunication Union
JM	Joint Model
JSVM	Joint Scalable Video Model
JVT	Joint Video Team
MB	Macroblock
MC	Motion Compensation
MD	Mode Decision
ME	Motion Estimation
MPEG	Moving Pictures Experts Group
MSE	Mean Squared Error
MSVC	Microsoft Visual Studio
MV	Motion Vector
QCIF	Quarter Common Intermediate Format
QP	Quantization Parameter

PDA	Personal Digital Assistant
PSNR	Peak Signal to Noise Ratio
RD	Rate-Distortion
RDO	Rate-Distortion-Optimized
SAD	Sum of Absolute Differences
SSD	Sum of Squared Differences
UMHexagonS	The Hybrid Unsymmetrical-cross Multi-Hexagon-grid Search
VCEG	Video Coding Experts Group
WF	Wave-front

# Chapter 1

## Introduction

As we move further into the information age, the amount of information surrounding us is growing more than ever. With the latest technological advancements in electronics, chips now can carry millions of transistors and possess unprecedented computing abilities. However, the management of the growing data is an increasing concern. Moreover, the developments in communications and information theory have created different media through which data may be transmitted; and as the transmission and the storage of every single bit incurs a cost, developing efficient compression algorithms has gained significant importance. Data compression is the process of representing data in a more compact form for efficient transmission or storage (i.e. with fewer bits). Compression algorithms reduce the data size by recognizing and exploiting the redundancies present in data. Different data types exhibit different types of redundancies, hence the algorithms used in the compression of different data types also differ.

This chapter serves as an introduction to the research work presented in the thesis. The first section briefly reviews the video compression theory. The next section discusses the topic of computational complexity in video compression and formulates the problem addressed in this thesis. Then the contribution of this research work is summarized. The chapter concludes with an outline of the thesis.



**Figure 1.1** Illustration of spatial correlations within a frame.

## 1.1 Video Compression Basics

Video technology has significantly evolved over the years. With digital video growing more popular, the immense volume of data present in video renders its storage and transmission very expensive as bandwidth is a very valuable commodity in today's world. However, video contains copious amounts of redundancy and video compression techniques are based on the removal of two types of redundancies:

- Temporal redundancy
- Spatial redundancy

A video is in fact a group of consecutively captured pictures. Each picture, within itself, exhibits spatial redundancy to a certain extent. Fig. 1.1 shows an arbitrary frame from a video sequence. Within the frame, there are highly detailed regions but also some homogeneous regions that exhibit similar content. The circled regions indicate such homogeneous regions. In addition to spatial redundancy, temporally consecutive frames tend to be highly correlated. As Fig. 1.2 displays, there are usually small changes between two pictures (unless there is a scene change). Video compression algorithms operate on the premise of removing both the temporal and the spatial redundancies while maintaining an acceptable level of visual quality; and compress a video sequence into a bitstream for condensed storage and efficient transmission [1].



**Figure 1.2** Illustration of temporal correlations between two consecutive frames.

## 1.2 Problem of Computational Complexity in Video Compression

Over the years, video compression algorithms have improved in their compression efficiency. H.264/AVC is the state-of-the-art international video coding standard developed by the *Joint Video Team* (JVT) [2]. JVT is a joint workforce established by two study groups, ITU-T Video Coding Experts Group (VCEG) and ISO/IEC Moving Pictures Experts Group (MPEG), with the vision of finalizing the new video coding standards.

H.264/AVC adopts various highly advanced and flexible encoding techniques to achieve significantly higher compression efficiency compared to previous standards. The high performance of the new standard renders it the prevalent compression algorithm in many applications and platforms. However, the encoding techniques employed in the standard come with a price of increased computational complexity. Although designs that are capable of performing such complex functions are realizable with the improved computational power of the IC chips, the increased computational complexity poses a great problem particularly for the battery operated portable devices with limited processing capabilities such as mobile phones and PDAs.

The design of computational complexity control algorithms for video coding has become a highly active research field. A complexity control structure should comprise of two distinct components, a low complexity component and a complexity scalable component that can adapt the computational complexity of the system and operate under variable resource constraints. In the recent years, different techniques have been developed which are either low complexity or complexity scalable but only a few that joins the two notions together.

### 1.3 Thesis Contribution

The purpose of this research was four-fold:

1. Understanding and modeling the encoder complexity of H.264/AVC
2. Developing complexity reduction algorithms for H.264/AVC encoding
3. Devising a novel complexity scalable H.264/AVC compliant encoding framework
4. Assembling a full-fledged H.264/AVC encoder by incorporating the complexity reduction algorithms in the complexity scalable framework

The proposed encoder embodies both of the two aforementioned design principles. The complexity scalability is maintained efficiently by a single control parameter. Results show that the complexity modeling of the proposed encoder is more precise than that of a similar work by other researchers and that the computational complexity is reduced significantly. It is also shown that the complexity scalable nature of the scheme provides multiple modes of operation that would help the encoder persevere through different resource constraints.

### 1.4 Thesis Overview

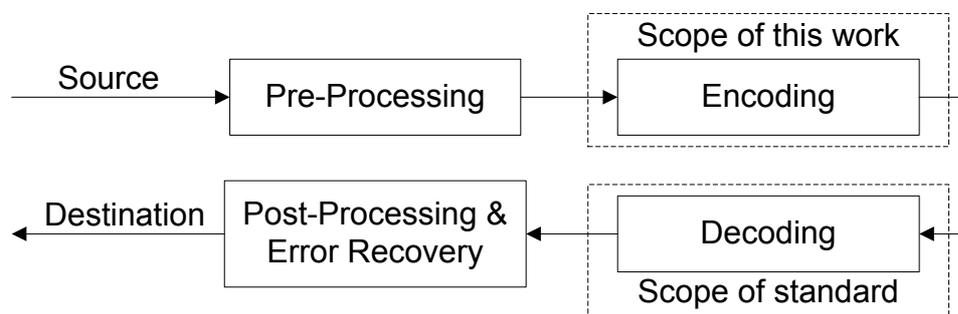
The remaining chapters are organized as follows: Chapter 2 introduces the H.264/AVC video coding standard and reviews the literature about the techniques that were developed for complexity reduction and scalability of the H.264/AVC encoder. Chapter 3 describes the testing methodology and the testing tools used in this work. Chapters 4 analyzes the mode decision complexity of an H.264/AVC encoder; it proposes a novel rate-distortion cost prediction method and two complexity reduction algorithms which use the prediction

method. Chapter 5 presents a novel complexity scalable encoding framework and installs the developed complexity reduction techniques into this framework. In Chapters 4 and 5, following the presentation of a new algorithm, the corresponding simulation results and their discussion are also provided. Finally, Chapter 6 concludes the thesis by summarizing the contributions and by investigating some further extensions of this research work.

## Chapter 2

# Overview and Complexity Analysis of H.264/AVC

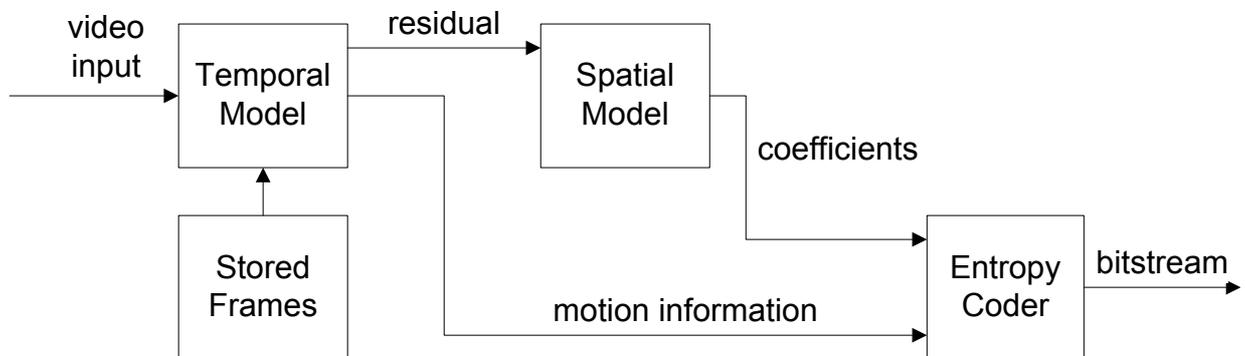
This chapter breaks down the key aspects of the H.264/AVC video coding standard and reviews some of the existing complexity reduction algorithms and the few complexity scalable frameworks available. The first section of the chapter provides a synopsis of the generic video encoding steps. The next section elaborates on those steps as they are realized in the scope of H.264/AVC standard. The final section explores previous research on resolving the complexity issues of H.264/AVC and discusses a collection of techniques proposed by other researchers.



**Figure 2.1** Scopes of the H.264/AVC standard and this work.

Similar to prior standards, H.264/AVC standardizes only the decoding process by imposing restrictions on the bitstream and syntax, as depicted in Fig. 2.1. Such standardization allows designers maximum freedom in encoder implementation and guarantees that

every conforming decoder will produce similar output when given an H.264 compliant bitstream [2]. The work presented in this text focuses on the encoder side. Thus, a technical overview of only the H.264 encoding process will be presented. However, for the sake of better understanding, a short introduction to the generic encoder structure will precede the in-depth analysis of the H.264 encoder.



**Figure 2.2** The block diagram of a generic hybrid video encoder.

## 2.1 Video Encoding Concepts

Since the early 1990s, the major video coding standards have evolved around the same hybrid encoding structure which incorporates a block-based predictive coding stage that removes the temporal redundancies and a transform-domain quantization stage that removes the spatial redundancies. This structure is referred to as a hybrid structure, due to its temporal/spatial duality. The list below gives the definitions of commonly used terms in block-based hybrid video encoding. The bullet points are organized in an order of progression that helps the understand the basic of a generic video encoding system. The list covers the fundamentals of the whole system and the terms pertinent to the thesis are covered in detail in the proceeding sections.

- **Pixel:** A pixel is the smallest discrete element in a digital picture. Typically, a picture is formed by a rectangular array of pixels.
- **Luminance Component:** A luminance (luma) component represents the brightness in an image. Typically, there is a luma component for each pixel.

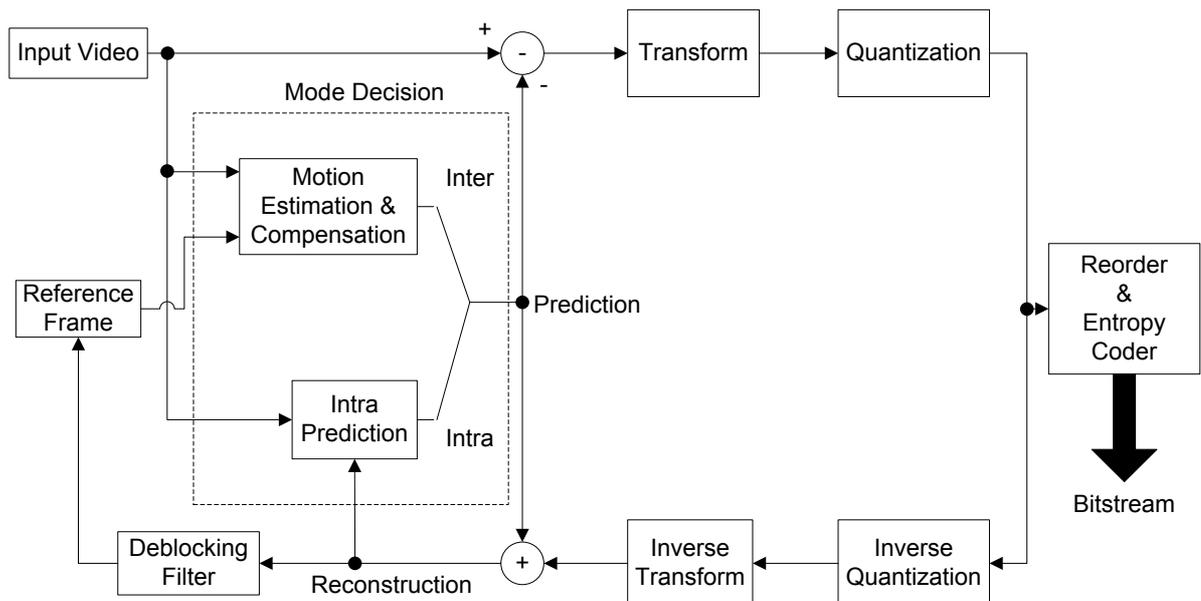
- **Chrominance Components:** A pair of chrominance (chroma) components represents the blue and the red colors in an image. Typically, there is a pair of chroma components for every four pixels.
- **Sample:** A sample is another substitute term for a luma or a chroma component.
- **Sampling Format:** Sampling formats refer to the different ratios of luma and chroma samples per pixel. Typically, in H.264/AVC encoder implementations the default sampling format is 4:2:0 which is also true for this thesis. 4:2:0 sampling format refers to the case where there is a luma sample for each pixel and a chroma sample pair for every four pixels.
- **Block:** A block is an  $M \times N$  matrix of samples.
- **Macroblock:** A macroblock is a  $16 \times 16$  matrix of pixels. A macroblock may be divided into smaller blocks which are also referred to as its partitions.
- **Frame:** A frame is an array of pixels representing a single time instant of a video sequence. In this thesis, the terms ‘frame’ and ‘picture’ are used interchangeably.
- **Motion Estimation:** Motion estimation is the process of finding the block with the most similar content in the previously coded frame(s) for a particular block in the current frame. This block is generally referred to as the ‘*best matching block*’ which is also the prediction of the current block.
- **Motion Compensation:** Motion compensation is the process of computing the difference between a block and its prediction. This difference is referred to as the residue or the residual block.
- **Motion Vector:** Motion vectors are vectors indicating the offset between a macroblock and its prediction. A macroblock may have more than one vector as the macroblock may be consisting of multiple partitions. These vectors represent the temporal motion of the macroblock between frames.
- **Transform:** The transform is the process of translating a residual block from the pixel domain to the frequency domain. The output of the transform domain is a set of transform coefficients.

- **Quantization:** Quantization is the process of mapping an input with  $X$  possible values to an output with  $Y$  possible values, where  $Y \leq X$ . The output has a smaller range of possible values than the input.
- **Entropy Coding:** Entropy coding is a lossless data compression scheme. In video encoding, the output of the entropy coder is the compressed bit-stream.

Fig. 2.2 illustrates a high level block diagram of a typical hybrid video encoder. The different encoding elements can be grouped into three main functional units [1]:

- Temporal model
- Spatial model
- Entropy coder

The temporal model is responsible for removing the temporal redundancies between consecutive frames. Similarly, the spatial model is responsible for removing the spatial redundancies within a frame and the entropy coder is responsible for removing any remaining statistical redundancies in the data.



**Figure 2.3** The block diagram of a typical H.264 Encoder.

## 2.2 H264/AVC Encoder

The previous section provided a synopsis of the hybrid video encoding framework and broke it down into three black-box functional units. This section will elaborate on those functional units as they manifest themselves in the H.264/AVC encoding framework.

Similar to its predecessors, H.264/AVC is also based on the hybrid video encoding framework and conforms with the hybrid encoder structure. The block diagram of an H.264/AVC encoder is illustrated in Fig. 2.3. As seen in the figure, the encoder consists of two distinct data flow paths.



**Figure 2.4** An arbitrary frame divided into macroblocks each of size  $16 \times 16$  pixels.

The forward data path is the encoding process of a macroblock. A prediction of the macroblock is constructed either using an INTER prediction through motion estimation and motion compensation or using an INTRA prediction. The decision making between the INTER and INTRA prediction options is identified by the *Mode Decision* block. The *Motion Estimation & Compensation* block and the *Residue Subtraction* make up the temporal model. The *INTRA Prediction* block together with the *Transform* and the *Quantization*

blocks make up the spatial model. The *Reorder & Entropy Coder* block makes up the entropy coder alone. Each of these blocks will be explained thoroughly in the following subsections.

The backward data path is the reconstruction process of the encoded data. After a frame is encoded, it is also decoded and then reconstructed again by the encoder. These reconstructed frames are stored and used in the prediction of the proceeding frames as references. In hybrid encoders, the backward data path is crucial as it guarantees that both the encoder and the decoder use the same reconstructed frame for prediction. The backward data path was omitted in Fig. 2.2 for simplicity.

### 2.2.1 Mode Decision

Video frames are partitioned into fixed-sized macroblocks (MBs) that comprise an area of  $16 \times 16$  pixels as shown in Fig. 2.4. The encoder processes each input frame in units of macroblocks and constructs a prediction for each macroblock based on previously coded data. The previously coded data can be either from the current frame or from other frames. A macroblock that is predicted from one or more MBs in other frames is referred to as an INTER predicted MB. A macroblock that is predicted from one or more MBs in the current frame is referred to as an INTRA predicted MB.

H.264/AVC standard offers more INTER and INTRA prediction methods than any previous standard did. A ‘coding mode’ or simply ‘mode’ is a synonym for a prediction method. The abundance of coding modes enables the encoder to construct a more accurate prediction which yields significantly improved compression efficiency compared to previous standards.

#### INTRA Prediction

If a frame contains new information, e.g. a new scene, that was not present in the previous frame, the temporal correlation becomes minimal. In such scenarios predicting from the spatially neighboring pixels instead of the temporally neighboring pixels is more desirable because the probability of having a spatially correlated region as opposed to a temporally correlated one is higher. Hence, the encoder chooses INTRA prediction over INTER prediction.

H.264/AVC offers two types of INTRA coding, namely the INTRA  $4 \times 4$  modes (I- $4 \times 4$ )

and the INTRA  $16 \times 16$  mode (I<sub>16</sub> $\times 16$ ). For an I<sub>4</sub> $\times 4$  mode, an independent prediction is formed for each  $4 \times 4$  block partition of a macroblock; whereas for an I<sub>16</sub> $\times 16$  mode, a single prediction is formed for the entire macroblock [2]. The former is more suitable for coding of the parts with significant detail and the latter is more suitable for coding of the smoother parts with less detail.

For each  $4 \times 4$  partition, one of the nine INTRA  $4 \times 4$  coding modes may be chosen as illustrated in Fig. 2.5. The 16  $4 \times 4$  partitions of a MB labeled as ‘a’ to ‘p’ are predicted from the previously encoded and decoded partitions of the adjacent MBs [1]. For instance, Mode 0 creates a prediction by copying only the samples from above (vertical) while Mode 1 creates a prediction by copying only the samples from left (horizontal) and Mode 2 by averaging the adjacent samples (DC).

For the entire macroblock, one of the four INTRA  $16 \times 16$  coding modes may be chosen as illustrated in Fig. 2.6. Similar to INTRA  $4 \times 4$  modes, Mode 0 is vertical prediction, Mode 1 is horizontal prediction and Mode 2 is DC prediction with the distinction that in INTRA  $16 \times 16$ , 16  $4 \times 4$  partitions are used on each side to predict.

## INTER Prediction

Video coding techniques achieve most of the compression by exploiting temporal redundancies and constructing INTER-frame predictions, i.e. predicting the current frame from other previously encoded frames. H.264/AVC offers a vast number of coding options for INTER prediction.

Each macroblock has two lists of reference pictures to predict from, *list 0* and *list 1* [2]. The former stores temporally preceding frames and the latter stores temporally proceeding frames. The standard allows the coding order of the frames to be different than the display order of the frames. Therefore, frames can be predicted using the references in *list 1* as long as the application permits (i.e. non-real-time encoding) because encoding the frames ahead of time entails a certain amount of delay which is tolerable only if the encoding is not real-time. Depending on the reference list used, INTER macroblocks are categorized in two groups. A P-macroblock (**P***redicted*) can use only *list 0* references whereas a B-macroblock (**B***i*-*predicted*) can use both *list 0* and *list 1* references [2].

In INTER prediction, the prediction of a MB is constructed upon searching for the best matching block of pixels in the reference frame and compensating for the motion between

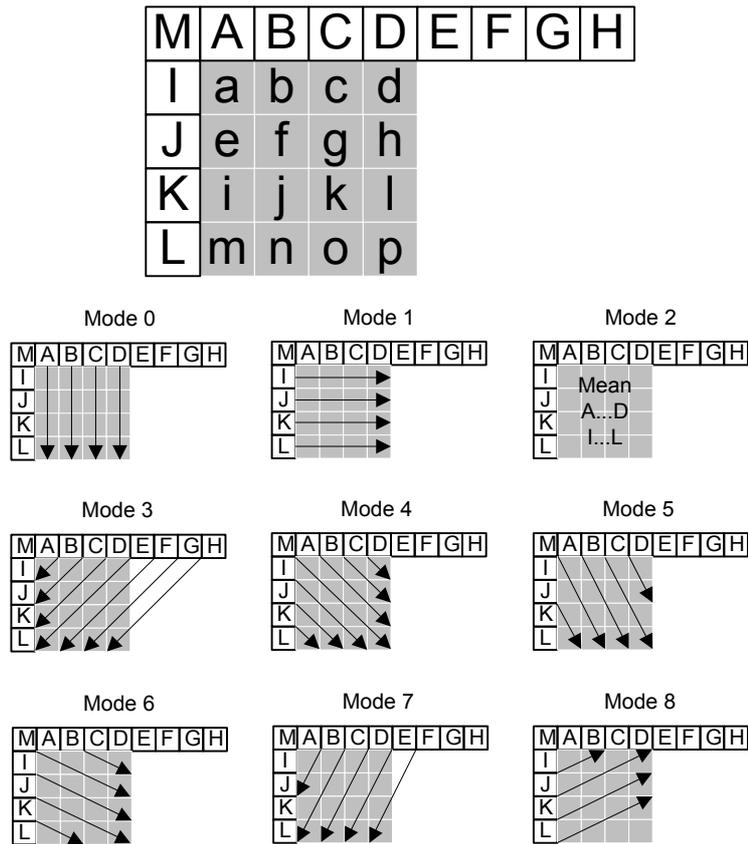


Figure 2.5 The supported INTRA\_4 × 4 modes.

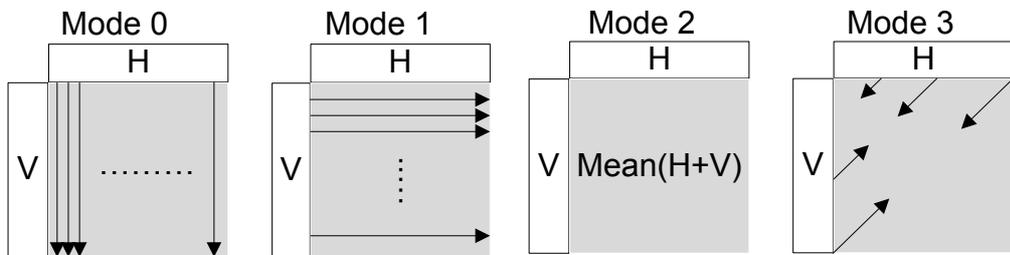


Figure 2.6 The supported INTRA\_16 × 16 modes.

the MB and the matching block. Similar to INTRA prediction, H.264/AVC provides different INTER coding modes where each coding mode corresponds to a specific partitioning of the macroblock for motion search (estimation) and compensation. The available MB partitions in H.264/AVC are one  $16 \times 16$  block, two  $16 \times 8$  blocks, two  $8 \times 16$  blocks, and four  $8 \times 8$  blocks, i.e. P\_8  $\times$  8 mode. If P\_8  $\times$  8 is selected, the corresponding  $8 \times 8$  blocks can be further divided into two  $8 \times 4$  blocks, two  $4 \times 8$  blocks or four  $4 \times 4$  blocks [1]. The INTER prediction modes are illustrated in Fig. 2.7.

The plethora of INTER coding modes (each with different MB partition sizes) supported by H.264/AVC complicates the macroblock partition decision process. In the H.264/AVC encoder implementations, a Lagrangian rate-distortion-optimized (RDO) method is commonly used to select between the different INTER modes [3]. Per this method, for each coding mode, the encoder searches an area in the reference frame to find a best matching block to each partition. It compares an  $M \times N$  block with the  $M \times N$  blocks in the reference region and picks the one that minimizes an optimized Lagrangian cost function of *distortion* and *rate* [4]. Distortion is a quantification of the difference between the current  $M \times N$  block and the candidate region, and rate is the number of bits needed to represent the motion vector which holds the offset between the current block and the candidate region. The equation is as follows [4]:

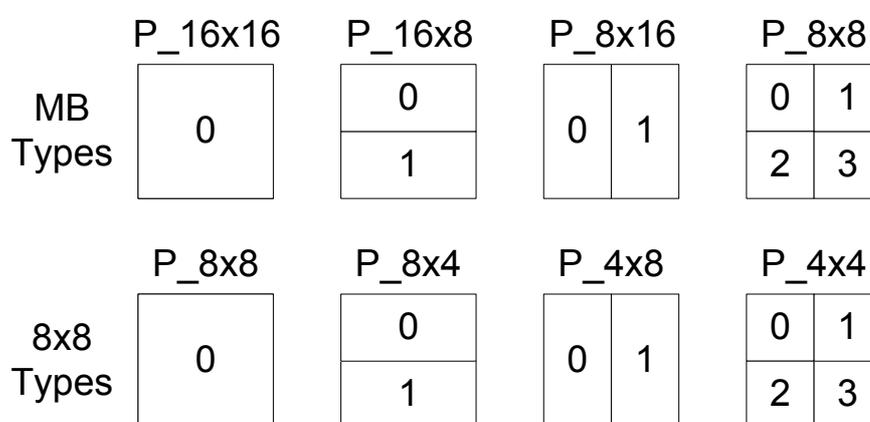
$$J_{MOTION} = D_{MOTION} + \lambda_{MOTION} \cdot R_{MOTION} \quad (2.1)$$

$J_{MOTION}$  is the Lagrangian motion cost.  $D_{MOTION}$  is either the sum of absolute differences (SAD) or sum of squared differences (SSD) (depending on the configuration options) between the current block and the candidate prediction block and  $R_{MOTION}$  is the bit-rate representing motion information and is calculated through a look-up table.  $\lambda$  is the Lagrangian multiplier and a function of the quantization parameter (QP). QP is a scaling index which controls the quantization step size and adjusts the visual quality. The Lagrangian multiplier is computed as follows [4]:

$$\lambda_{MOTION} = \sqrt[2]{0.85 \cdot 2^{(QP-12)/3}} \quad (2.2)$$

For instance, for P\_16  $\times$  16 mode the MB is not partitioned into smaller blocks; therefore, the  $16 \times 16$  block in the reference frame that minimizes  $J_{MOTION}$  and the corresponding motion vector representing the offset are used to construct the prediction. The search for

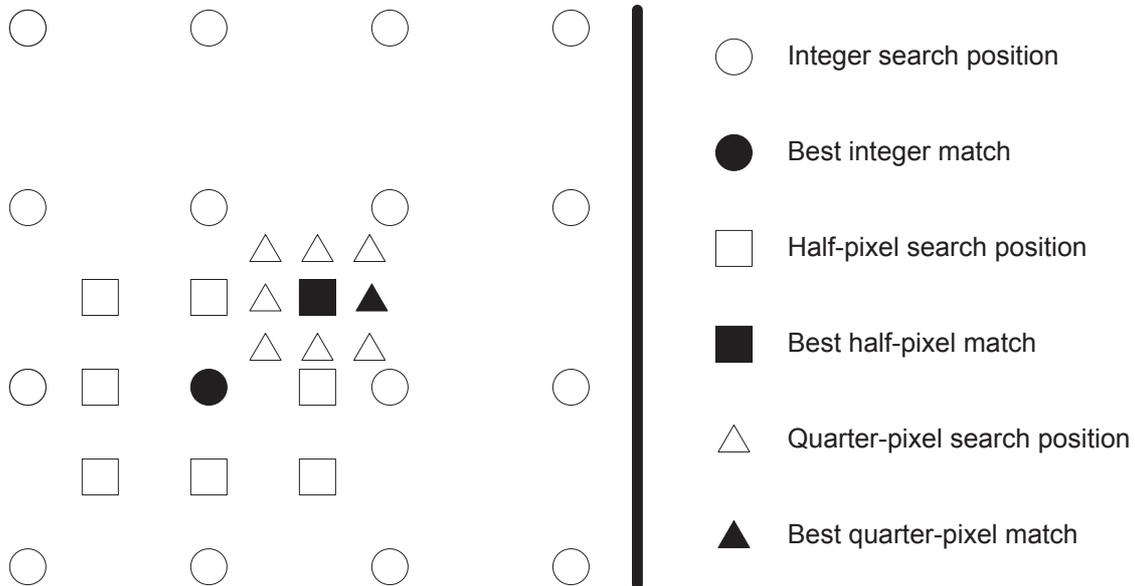
the best matching block is referred to as the *motion estimation* [1]. For  $P_{16 \times 8}$  mode the MB is partitioned into two  $16 \times 8$  blocks hence motion estimation is performed for each block separately. Once the best matching blocks for a MB are found, the motion between the current blocks and the reference blocks are compensated to form the prediction which is subtracted from the original MB and the residual is obtained. This process is referred to as the *motion compensation* [1].



**Figure 2.7** The supported INTER prediction modes.

H.264 supports sub-pixel motion estimation and compensation which improves the prediction accuracy. The reference block is actually constructed by creating new pixels through interpolation inside a block of real pixels. Half-pixel and quarter-pixel accuracies are supported by the standard. For instance, in quarter-pixel accuracy, motion estimation happens in three stages. In the first stage, the best matching block is found on the integer sample pixels. In the second stage, the half-pixel positions neighboring the best match is searched for improvement. If a better matching block is found through half-pixels, then in the third stage, the quarter-pixels are searched for further improvement. The interpolation process is illustrated in Fig. 2.8.

In addition to the motion-compensated modes, H.264 offers another mode known as the SKIP mode [2]. For this coding mode, neither motion estimation is done nor any data is transmitted. Simply a motion vector is predicted by averaging the motion vectors of the adjacent MBs and the region pointed by the vector is the actual reconstruction. This mode is used to predict large areas with no change or constant motion such as the background.



**Figure 2.8** The illustration of half and quarter pixel interpolation from integer pixel positions.

### Rate Distortion Optimized (RDO) Mode Decision

Typically, an H.264 encoder offers a total of 21 coding modes for a macroblock [2]:

- Nine INTRA\_4 × 4 modes
- Four INTRA\_16 × 16 modes
- Seven INTER modes with different partition sizes
- One SKIP mode

The encoder is expected to choose one of these coding modes for each macroblock such that for a given bit-rate constraint the visual quality is maximized. To accomplish this task, the encoder needs a reliable mode decision method. Similar to the INTER mode selection process, H.264/AVC encoder implementations typically employ a so-called *Rate-Distortion-Optimized Mode Decision* process [3]. For each coding mode, the process forms a prediction of the MB and computes the residual which is then frequency transformed and quantized. Then, the MB is decoded and reconstructed: the quantized transform coefficients are inverse quantized and inverse transformed to obtain the residual block which

is then added to the prediction block to reconstruct the MB. Using the reconstruction and the transform coefficients, the following Lagrangian cost function is computed [4]:

$$J_{MODE} = D_{MODE} + \lambda_{MODE} \cdot R_{MODE} \quad (2.3)$$

The above equation seems exactly the same as the one used in the motion estimation yet there are few crucial differences.  $D_{MODE}$  is the distortion representing the difference between the original MB and its reconstruction (typically SSD).  $R_{MODE}$  is the bit-rate calculated from the output of the entropy coder which represents the total number of bits required to hold the residual MB and the accompanying motion information in the bitstream. The Lagrangian multiplier,  $\lambda$  is again a function of the QP, yet different:

$$\lambda_{MODE} = 0.85 \cdot 2^{(QP-12)/3} \quad (2.4)$$

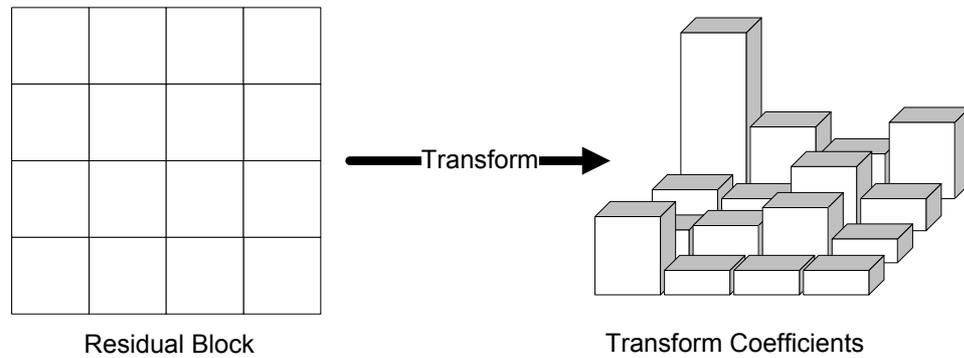
The Lagrangian optimization used in RDO mode decision provides a method of minimizing distortion subject to a rate constraint which coincides well with the goal of a video compression technique as stated in the previous chapter. Since the process is repeated for every possible coding mode, it guarantees that the optimal mode will be picked.

### 2.2.2 Transform

H.264/AVC uses an integer transform (IT) [5], depicted simply in Fig. 2.9. As opposed to the Discrete Cosine Transform (DCT) [6] that is commonly used in the previous standards, IT is carried out using integer arithmetics and can be implemented by additions and shifts; therefore, it is less complex than DCT. Since IT does not use any floating point arithmetics, any possible mismatch between the forward and reverse transform operations are eliminated. The data in the transform domain is de-correlated as much as possible to make the independent coding of separate samples possible [2].

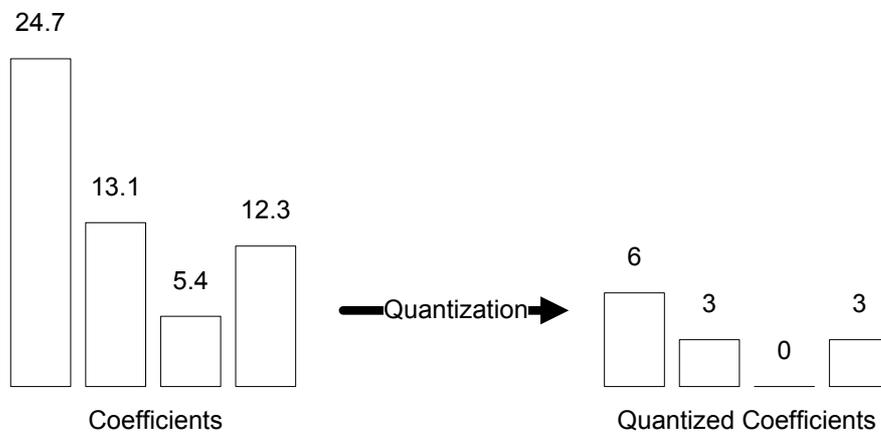
### 2.2.3 Quantization

Quantization reduces the amount of spatial detail and is the sole lossy operation in the entire video coding process. It exploits the fact that the human eye's perception does not detect subtle details. In H.264/AVC, the quantization parameter can take 52 values [2]. An increment of six in QP doubles the quantization step size [5]. A high QP value will



**Figure 2.9** Transform of a block of residual pixels into a set of transform coefficients.

increase the number of zero transform coefficients which results in coarser visual quality. The quantization of the coefficients is illustrated in Fig. 2.10.



**Figure 2.10** Quantization of a set of transform coefficients.

### 2.2.4 Entropy Coder

Prior to being fed to the entropy coder, the quantized coefficients are reordered in a particular order such that zero coefficients are clustered consecutively. The ordering helps the entropy coder as it converts a series of symbols representing the video elements in a compressed bitstream by mapping sequences of symbols to codewords based on the data statistics. Entropy coding is a lossless operation that exploits the statistical redundancies.

H.264/AVC standard offers two entropy coding modes, Context Adaptive Variable Length Coding (CAVLC) and Context Adaptive Binary Arithmetic Coding (CABAC) [2].

### 2.2.5 Deblocking Filter

The processing of frames in units of macroblocks in block-based coding assumes that the motion in a video sequence can be approximated by grouping of pixels in rectangular blocks. Although this assumption holds in general, it fails along the block edges and results in visual artifacts, which disturb the observer. The deblocking filter is designed to remove these artifacts and smoothen the samples along the block edges [2].

## 2.3 Encoder Complexity Management Algorithms

Video compression algorithms are comprised of a series of highly complex encoding tools including motion estimation/compensation, transform/quantization and reorder/entropy coder. Prior to H.264/AVC, the video coding standards had more or less an equal complexity distribution over their employed coding tools [7]. However, due to the increased number of supported coding modes in H.264/AVC, motion estimation and mode selection processes are more computationally intensive in comparison to transform/quantization and reorder/entropy coder. Together the processes may consume up to 90 to 95 percent of the total encoding time. Hence complexity management research for H.264/AVC has concentrated on developing:

- Low complexity motion estimation algorithms
- Low complexity mode decision algorithms
- Complexity scalable encoding algorithms

### 2.3.1 Low Complexity Motion Estimation Algorithms

For motion estimation, the *Full Search Algorithm* (FSA) is the most straightforward search method for block matching. It examines all the pixel positions and finds the motion vector pointing to the best matching block. Although the optimal solution to motion estimation is guaranteed, FSA can be computationally intensive as the search range increases. For

instance, if the search range for block matching in the reference picture is a square with  $w$  pixels in size for each direction and with a step size of one pixel, there exists a total of  $(2w + 1)^2$  possible motion vectors. As this operation is repeated for every block in a sequence, the total number of motion search operations is very large.

In order to relieve the encoder of the computational burden of FSA, many fast motion search algorithms have been proposed. Such algorithms decrease the computational complexity of the motion estimation operation by reducing the number of candidate motion vectors. Some of the earlier works in the field include three-step search [8], two-dimensional logarithmic search [9], cross-search [10] and diamond search [11]. Each algorithm uses a different method to eliminate the motion vectors that are likely to yield a large residue. For instance, in two-dimensional logarithmic search, at each search iterations, four different blocks are tried for a block with coordinates  $(x, y)$ . The search starts centered around the following four pixel locations:  $S$  pixels north,  $S$  pixels south,  $S$  pixels east, and  $S$  pixels west of  $(x, y)$ , where  $S$  is the search step size. After each iteration, if the central pixel location prevails among the four surrounding pixels, the search step is halved for the next iteration. This process is carried out until the search step is reduced to one. In three step search, eight surrounding locations are tested instead of four and the search step is proportional to the search window size. If the search window size is  $(2^N - 1)$ , then the initial search step size is set to  $S^{N-1}$ ; and similar to the two-dimensional logarithmic search, the algorithm carries out until search step is equal to one.

Fast motion estimation algorithms have been developed specifically for H.264/AVC. In fact, *The Hybrid Unsymmetrical-cross Multi-Hexagon-grid Search* (UMHexagonS) [12] is adopted in the H.264/AVC reference test model software [13] (see Section 3.1). UMHexagonS exploits previous motion information such as the motion vectors of the neighboring blocks, motion vectors for the INTER coding modes with larger partitions than the current coding mode and the motion vectors of the co-located block in the previous frame for current coding mode. Using such information, UMHexagonS predicts an initial minimum cost integer-pixel motion vector and a sub-pixel motion vector. An integer-pixel motion search is performed around the integer-pixel motion vector. Different search shapes (unsymmetrical cross, local square full search, and a hexagon based search) are used in order to prevent stalling in of the local minimums. After the integer-pixel motion search, a sub-pixel motion search is performed around the sub-pixel motion vector.

Results show that UMHexagonS saves from 41 to 67 percent of the total encoding time

compared with FSA while achieving the same rate-distortion performance. Yi, Zhang, Ling and Shang extend UMHexagonS by using a simpler motion vector prediction method (excluding the motion information from the co-located block in the previous frame) and by replacing the local square full search with a hexagonal search [14]. The authors claim further 40 to 60 percent encoding time savings [12].

Another very popular fast motion search algorithm is the *Enhanced Predictive Zonal Search* (EPZS) [15]. EPZS mainly comprises of three steps. Similar to UMHexagonS, it starts by selecting a best motion vector prediction from the available predictors. Next, the adaptive early termination terminates the search if a stop criterion is satisfied. Finally, if the early termination criterion is not satisfied, motion estimation is further refined around the best predictor and iterated to improve the final prediction. Results show total encoding time savings ranging from 30 to 50 percent [15].

### 2.3.2 Low Complexity Mode Decision

In addition to the low complexity motion estimation algorithms, there has been a great amount of research work directed into developing fast macroblock mode decision algorithms. Such algorithms aim to alleviate the computationally intensive mode decision process. H.264/AVC encoders in particular need to perform a computationally expensive mode selection operation because the new standard supports a larger number of macroblock coding modes than any previous standard.

Fast mode decision algorithms for both INTRA mode decision [16] and INTER mode decision [17] have been proposed by the same authors and have been incorporated in the H.264/AVC reference test model software [13]. Both methods exploit the local characteristics of the current block. Specifically, using the local edge directional information obtained from the edge direction histogram along the INTRA prediction directions, only the more probable INTRA modes are favored. Results presented by the authors indicate average coding time savings between 20 and 30 percent while the impacts on rate and distortion are less than two percent increase and 0.1dB decrease respectively [16]. The fast INTER mode decision algorithm works similar to its INTRA counterpart. It uses the same edge detection technique to recognize the homogeneous regions and the difference between the current block and the co-located block in the reference frame to recognize the stationary regions. Based on these analysis, the algorithm zeros in on the more probable INTER

prediction block sizes. For instance, if a region is detected to be both homogeneous and stationary, then the  $P_{16 \times 16}$  mode is chosen without any further mode evaluation. From the experimental results, the fast INTER mode decision is shown to reduce the encoding time by 30 percent on average while the impacts on rate and distortion are again less than two percent increase and 0.1dB decrease respectively [17].

Jeon and Lee propose a selective INTRA mode decision algorithm where they put forward the hypothesis that an INTRA prediction mode is likely to be chosen for a block if the block is more correlated with its previously encoded spatial neighbors than with its previously encoded temporal neighbors [18]. Following this hypothesis, their proposed method restricts the INTRA prediction mode evaluation to only the blocks conforming with the hypothesis. The results indicate on average 30 percent encoding time saving with minimal loss in rate-distortion performance.

A very hot topic in the development of fast mode decision tools is the early SKIP detection (termination) algorithms. An early SKIP detection algorithm aims to detect the macroblocks that are likely to be coded in the SKIP mode and to avoid the evaluation of other modes. Since the evaluation of SKIP mode is computationally simple, it would only be wise not to evaluate the INTER or the INTRA prediction modes for such MBs. This is possible only if these MBs could be identified a priori. Jeon and Lee propose an early SKIP detection where a macroblock is coded in the SKIP mode only if the  $P_{16 \times 16}$  mode yields both a set of all zero transform coefficients and a motion vector pair that is equal to the predicted motion vector [18]. Their algorithm is also adopted in the H.264/AVC reference test model software [13] (see Section 3.1). Its performance depends on the motion characteristics of the input sequence and on the quantization parameter. The saved encoding is greater for the slow moving sequences and for the high quantization parameters. The savings are less than 50 percent with negligible impact on rate-distortion performance.

Rate-Distortion (RD) cost estimation is a scarcely investigated fast mode decision tool, still bearing a great potential. Kannangara et al. developed a RD cost estimation method for a macroblock using the distortion and the rate values of the co-located block in the reference frame. This estimated RD cost is used in their proposed early SKIP detection algorithm [19] such that the MBs for which the estimated RD cost is less than the SKIP mode RD cost, are coded in SKIP mode without any additional mode evaluation. The method achieves on average 60 percent encoding time savings surpassing the early SKIP detection method by Jeon and Lee [18]. However the rate distortion performance is significantly infe-

rior as the rate-distortion-optimized mode decision is disabled. Liao, Yang and Sun propose another RD cost estimation for both INTER and INTRA prediction modes by exploiting the statistical information from the transform coefficients. However the encoding time savings are only after a residual block is integer transformed; hence computational time spent in neither the motion estimation nor the mode decision processes is really altered.

### 2.3.3 Complexity Scalable Algorithms

While the low complexity algorithms discussed so far reduce the computational complexity of the encoder, they lack the ability to adjust the overall encoder complexity on the fly, i.e. during the encoding process. Such ability is critical when the encoder is expected to persevere through a resource-constrained situation and complete the encoding at the expense of reduced performance. This requires some kind of a flexibility that can only be provided with a complexity scalable encoding framework. A complexity scalable encoder would reduce its operational complexity and sustain the battery power for longer durations in a limited battery power scenario or, in a limited computational resource scenario, allow the operating system to distribute the processing resources over the running tasks more judiciously. In the past, a few complexity scalable algorithms have been proposed that extend the low complexity motion estimation algorithms discussed earlier [20, 21, 22]. These methods introduce control parameters that vary the search termination criteria of their fast motion estimation algorithms; thus adjusting the computational complexity.

More recently, Kannangara, Richardson and Miller proposed a novel complexity management technique for H.264/AVC encoding. The technique comprises of two control levels: a frame-level control and a per-frame level control [5]. The frame-level control algorithm initially calculates a target coding time for the next video frame based on the overall coding delay and on the target frame rate. If the coding delay hinders the target frame rate, the frame is dropped. If the frame is kept, the decision power is passed to the per-frame level control algorithm which manages the computational complexity within the frame. It uses their previous work on early SKIP detection [19] and varies the threshold that is used to identify the skipped macroblocks. The variable block-skipping guarantees that the frame is encoded within the target encoding time. The advantage of this method over the other complexity scalable methods is its per-frame complexity control. Complexity scalable methods generally control the complexity only at a frame level by dropping frames which

in turn leads to unwanted choppy perceptual video quality at the decoder. However, the disadvantage of their method is the fact that the highly beneficial rate-distortion-optimized mode decision is disabled in order to enable the employed early SKIP detection.

Tan, Lee, Thai and Rahardja have recently proposed a macroblock level complexity scalable H.264/AVC encoding framework [23]. The proposed method adopts the so-called ‘wave-front macroblock scheduling’ to provide the MB-level scalability. Macroblocks are grouped in wave-fronts (sets) such that the MBs in a wave-front do not rely on each other during INTRA prediction and can be encoded independently. Each wave-front is incrementally allocated some complexity until a predefined complexity budget is exhausted. The complexity of the proposed algorithm is controlled by a single parameter yet the modeling is rather crude and in need of refinement.

## 2.4 Summary

The chapter opened with a generic video encoder discussion building towards the H.264/AVC encoder. The H.264/AVC encoder steps were discussed in detail with their contribution to the overall encoder complexity. The complexity problem of the encoder was investigated and the previous research in the field of complexity management of the encoder was reviewed in three categories:

- Low complexity motion estimation algorithms
- Low complexity mode decision algorithms
- Complexity scalable encoding algorithms

The algorithms in the first two categories aim to reduce the computational complexity of motion estimation and mode decision processes respectively. While some of the algorithms achieve significant computational savings, they lack the scalability notion which is essential for sustaining the encoding process under volatile resource constraints. The algorithms of the third category seek to solve this problem. However, the majority of the existing complexity scalable frameworks has serious drawbacks, such as the absence of a concrete frame-level control algorithm, the disablement of the rate-distortion-optimized mode decision or the inaccurate encoder complexity modeling.

The work presented in this thesis seeks answers to these problems by proposing a singly-parameterized complexity scalable H.264/AVC encoding framework and two complexity reduction algorithms. The scalability is attained through the wave-front grouping of macroblocks and a judicious complexity allocation strategy over the macroblocks is employed. The encoder complexity is modeled accurately by the single control parameter. The complexity reduction algorithms make use of a novel rate-distortion cost prediction method to identify the modes of macroblocks at earlier stages of encoding. The next chapter will describe the experimental method used in the development and in the testing of the proposed algorithms.

## Chapter 3

# Experimental Method

This chapter describes the experimental method employed in this research work. The following sections explain the used test tools, testing methodology and performance evaluation methods.

### 3.1 Video Encoder Software

The Joint Scalable Video Model (JSVM) [8] and the Joint Model (JM) [9] are the publicly available H.264/AVC reference software encoders developed by JVT. Both encoders can be downloaded through the internet [24] and [25] respectively. JSVM is developed with the purpose of providing the additional support for the scalable extension of the standard [1] which is not in the scope of this work. The reference encoders are used globally by researchers to test new algorithms in the video coding community. The use of common reference software creates a fair platform for comparing different algorithms.

In this research work, JSVM version 9.19.3 is used for the implementation and the testing of the algorithms and JM version 17.0 is used for testing the early SKIP detection algorithm [18] adopted by the JM reference software. Since the two encoders have different architectures, only the complexity reduction percentages are compared while examining two algorithms that are implemented on different encoders. This avoids any unwanted architecture-related impact on the performance of an algorithm in test.

Both JSVM and JM encoders comply fully with the H.264/AVC standard and work in a similar fashion. It should be noted that neither of the encoders exploits any special

instructions of any type of processor for the purposed of hardware acceleration. The input parameters are fed through a configuration file. The simulations discussed in the following chapters use some common configuration options. These common options are listed below:

- Input of 80 QCIF frames at 30 frames per second rate
- IPPP coding structure with one reference picture per frame
- Motion search using the Full Search Algorithm with a search range of 32 pixels (see Section 2.3.1)
- Sum of Absolute Differences (SAD) for full-pixel motion estimation
- Sum of Squared Differences (SSD) for sub-pixel motion estimation
- $4 \times 4$  luma transform
- CAVLC for the entropy coding option

After the encoding process, a collection of encoding statistics are provided along with the compressed bitstream by either outputting to the screen or writing into a text file. The statistics include the total number of coded bits, the bit-rate of the encoded bit-stream and the video quality for luma and chroma components in PSNR (discussed in Section 3.5.1).

### **3.2 Development Environment**

As JM and JSVM are developed in C and C++ programming languages respectively, Microsoft Visual Studio 2008 (Visual C++ compiler is commonly abbreviated as MSVC) is used as the Integrated Development Environment (IDE) to compile and build the code. The MSVC is also used to do the necessary modification on the codes that implement the algorithms in the encoder.

### **3.3 Test Platform**

Any difference in the test platform hardware could compromise the fair comparison of different algorithms. In this research work, all the tests are performed on the same test platform: a personal computer with the following specifications:

- Processor: Intel Core2 Duo CPU 2.33GHz
- Memory: 3.00GB
- Operating System: Microsoft Windows 7

It should be noted that during the course of this work, none of the encoders used in any of our simulations exploit any processor specific special instructions sets. Therefore, we expect that our results (in terms of relative performances of different encoders compared) should generalize well to other platforms and processors, such as a mobile platform with a much simpler processor.

### 3.4 Test Video Sequences

The video sequences used in the testing of the algorithms are chosen from the ITU-T test video collection. These test sequences are widely used by researchers and contain a variety of backgrounds and foregrounds, content detail, object and camera motion. The picture format of the sequences are ‘Quarter Common Intermediate Format’, commonly known as QCIF. QCIF videos are 176 pixels wide and 144 pixels tall. The list below describes the content of each sequence. Samples from each are shown in Fig. 3.1.

- Akiyo:** This video clip shows an anchorwoman from the waist up broadcasting news in a studio with a fixed background. The motion in the video is minimal and limited to the subtle head movements of the woman as she delivers the news.
- Football:** This video clip exhibits high motion as football players are diving on the ground. Although the recording camera is still, players running in and out of the scene creates a content change between frames.
- Foreman:** In this video clip, a construction worker is talking to a shaky hand-held camera with exaggerated gestures. Towards the end, the camera turns to the construction site as the worker points it out.
- NBA:** In this video clip, the camera follows the basketball as the attacking player dribbles the ball and shoots a jump shot while being confronted by an opposing team player. The camera follows the ball as the shot is missed and players try to get the rebounding ball.

- (e) **Silent:** This video clip, similar to Akiyo, shows a woman from the waist up as she does sign language. Both the camera and the background are still but the woman is highly active with her arms.
- (f) **Stefan:** In this video clip, the camera follows a tennis player as the ball is hit back and forth. The player is in constant motion and the camera focuses on the player by zooming in and out. Towards the end of the clip, the player runs to the net to get to the ball.

The sequences can be grouped into three categories in terms of their motion content. Akiyo and Silent are characterized to be slow moving sequences having still backgrounds and high temporal correlation between frames. Foreman and Stefan are faster sequences recorded with jittery hand-held cameras that focus on the body motion of the person in the video clip. Football and NBA are fast motion sequences with rapidly changing backgrounds and foregrounds as the objects continuously enter and leave the scenes. Additional test sequences have been used to test the algorithms for robustness: Carphone, Costguard, Container, Flower, Grasses, Miss-America, Mobile, Mtdt, and News. The majority of the above sequences can be downloaded online [10].

### 3.5 Testing Methodology and Performance Evaluation

The testing methodology followed in this work is illustrated by a block diagram in Fig. 3.2. For each test scenario, three performance indicators are considered: computational complexity, video quality and bit-rate. The computational complexity is measured during the encoding process. Any overhead introduced by a complexity management algorithm is also included in the complexity measurements. The video quality and the bit-rate values are obtained from the encoder outputs.

#### 3.5.1 Video Quality

The visual perception of a video is highly observer dependent. There are many factors affecting the outcome such as the occupation of the observer while viewing the video. This subjective nature makes it really difficult to assess the visual quality of a video. ITU-T has created a number of standard subjective video quality assessment methods for mobile



(a) Akiyo



(b) Football



(c) Foreman



(d) NBA

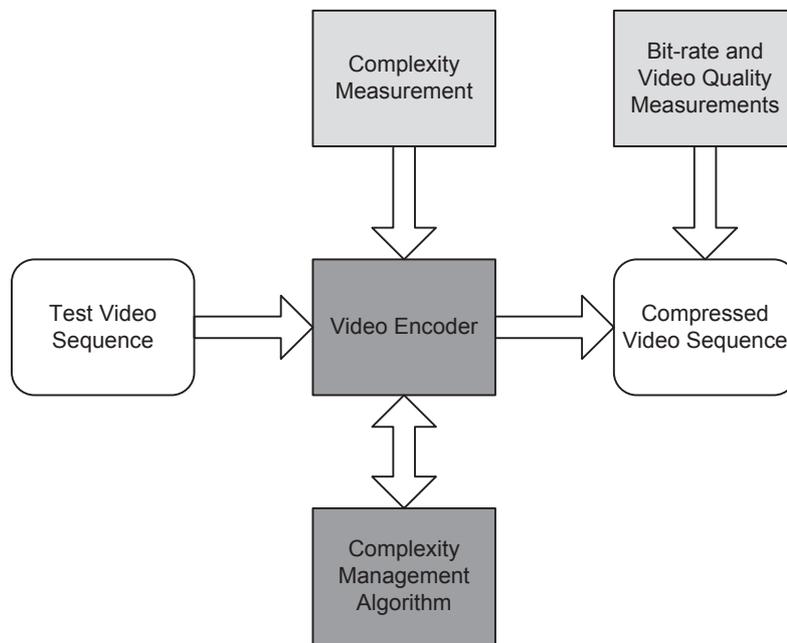


(e) Silent



(f) Stefan

**Figure 3.1** Snapshots of the 29<sup>th</sup> Frame of the 6 sequences Akiyo (a), Football (b), Foreman (c), NBA (d), Silent (e) and Stefan (f).



**Figure 3.2** The block diagram of a test scenario.

applications and for television pictures [11] and [15]. However, these methods tend to be highly time consuming and expensive because typically a large number of participants is required.

In this work, objective measurements are used to assess the video qualities. Such measurements differ in their methods to quantify the differences (distortion) between two videos. When compared to the subjective measurements, objectives are faster and more reliable as they can be easily repeated to get the same results. Peak Signal to Noise Ratio (PSNR) is the most commonly used objective measurement in the video coding community and is also the measurement used in this work. PSNR is expressed in terms of the logarithmic decibel (dB) scale and is calculated as follows:

$$PSNR = 10 \times \log_{10} \frac{2^b - 1}{MSE}, \quad (3.1)$$

where  $b$  is the number of bits per pixel and is typically equal to eight. MSE is the abbreviation for the Mean Squared Error between the original picture and the test picture where both pictures are  $M$  pixels wide and  $N$  pixels tall. MSE is computed as follows:

$$MSE = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N (p_o(i, j) - p_t(i, j))^2 \quad (3.2)$$

$p_o(i, j)$  denotes the pixel with coordinates  $(i, j)$  in the original picture and  $p_t(i, j)$  denotes the pixel with coordinates  $(i, j)$  in the test picture. MSE is the squared average of the pixel differences between the two pictures and can be a measure of distortion by itself. However, PSNR is usually preferred because the logarithmic decibel (dB) scale is a better representative of the wide dynamic range of a video signal. The PSNR of a complete video sequence is generally computed by averaging the PSNR values of all the pictures.

Throughout this thesis, two additional distortion measures are mentioned : The Sum of Absolute Differences (SAD) and The Sum of Squared Differences (SSD). These two methods are computed respectively as follows:

$$SAD = \sum_{i=1}^M \sum_{j=1}^N |p_o(i, j) - p_t(i, j)| \quad (3.3)$$

$$SSD = \sum_{i=1}^M \sum_{j=1}^N (p_o(i, j) - p_t(i, j))^2 \quad (3.4)$$

SAD and SSD are the less computationally means of quantifying the distortion between two pictures or two blocks. They are usually used in the Motion Estimation and Mode Decision processes respectively.

### 3.5.2 Bit-rate

The term ‘bit-rate’ refers to the number of bits processed per second. It is computed using the total number of bits encoded in the bit stream, the total number of coded frames and the target frame-rate (frames per second) as follows:

$$\text{Bit-rate} = \frac{\text{Frame-rate} \times \text{NumberOfBits}}{\text{NumberOfFrames}} \quad (3.5)$$

### 3.5.3 Computational Complexity

The algorithms discussed in this thesis aim to either reduce or control the computational complexity of the encoder. Therefore, it is critical that an accurate measurement of the

computational complexity is used. In software-only encoders, encoding time is a direct measurement of the encoder/algorithm complexity. In fact, the algorithms discussed in proceeding chapters are software-only algorithms tested on a single platform. Hence this thesis uses the ‘total encoding time’ to measure ‘the computational complexity of the encoder’ and the two terms are used interchangeably. For different encoders, the total time spent during the encoding of the sequences is recorded in milliseconds and used to compare different complexities.

### **3.6 Summary**

This chapter discussed the experimental method used in the remainder of the work. All the algorithms are software-only and tested using the same personal computer for fair comparison. Three performance evaluation tools are employed: PSNR, bit-rate and encoding time. PSNR and bit-rate are obtained from the reference JM and JSVM encoder outputs. Encoding time is recorded during the encoding process by timing the algorithm. The next chapter will introduce a novel rate-distortion cost prediction method and two new complexity reduction algorithms.

## Chapter 4

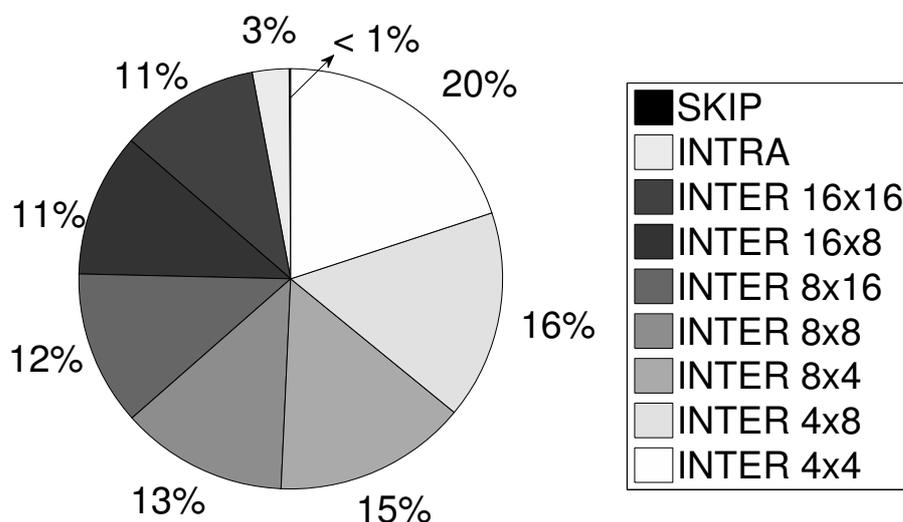
# Complexity Reduction Tools

This chapter introduces two novel complexity reduction algorithms. Prior to the presentation of the algorithms, an analysis of the mode decision complexity is presented through profiling the time consumption and the final mode decisions of the rate-distortion-optimized mode decision (MD) process. The profiling provides the distribution of the total MD processing time in percentages over individual mode trials together with the percentage occurrence for each mode. Inspired by this breakdown of the MD processing time, a novel rate-distortion (RD) cost prediction method is presented. This method forms the building block of the two complexity reduction algorithms.

### 4.1 Introduction

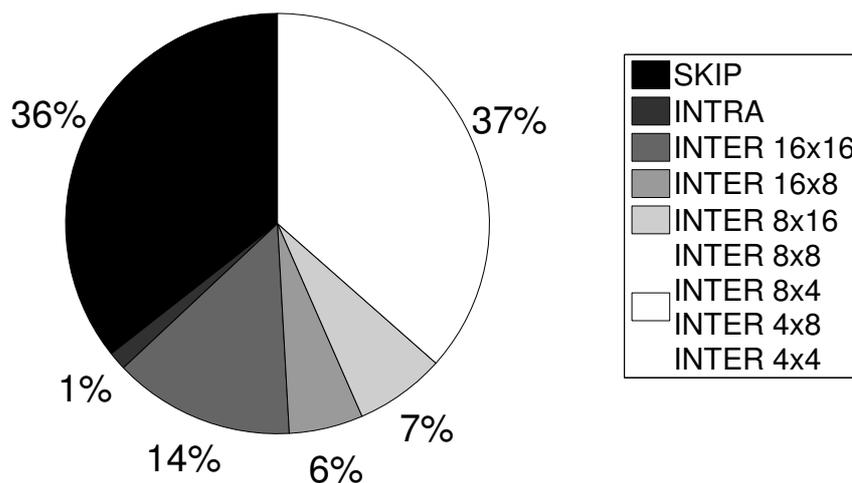
The increased number of available coding modes in H.264/AVC demands a more discriminating mode decision operation to be performed by the encoder such that all the coding modes are carefully analyzed to guarantee the optimal decision. The H.264/AVC encoder implementations typically utilize a rate-distortion-optimized (RDO) mode decision that ensures the optimal choice of mode for each macroblock (see Section 2.2.1). For each coding mode, the RDO mode decision carries out a number of operations, which examine both the rate and the distortion effects of coding the macroblock with the mode, and it calculates a joint rate-distortion penalty. From this point on, we will refer to this critical RD analysis of a mode as a “mode trial”. Each mode trial requires a number of complex operations to be performed; including motion estimation/compensation for INTER modes,

transform/quantization and reorder/entropy coding. Although the overall complexity of the RDO mode decision is large, the contributions of different mode trials to this complexity are not equal. To categorize the complexity of each mode, we profile the mode decision process. The time spent evaluating each mode and the outcome of the mode decision process for each macroblock are recorded for all the 15 test video sequences mentioned in Section 3.4. For each sequence, the quantization parameter is varied from 8 to 48 with a step size of 2.



**Figure 4.1** Average percentage distribution of computational time spent evaluating each mode. The numbers are rounded for the clarity of the graph.

Fig. 4.1 displays the average percentage distribution of different mode trials during the RDO mode decision process as a pie chart. It is seen that the INTER modes are the major complexity contributors, consuming together over 95 percent of the total time spent in the mode decision process. The highly complex nature of INTER modes is due to the exhaustive motion estimation operation performed for these modes. All INTRA modes consume just under three percent of the time in total and SKIP mode's consumption is negligible with less than one percent. Similarly, Fig. 4.2 displays the percentage distribution of macroblocks coded in each mode as a pie chart. It is seen that more than 60 percent of the macroblocks are coded in one of the seven INTER modes and more than 35 percent of the macroblocks are coded in SKIP mode. INTRA modes are preferred for only two percent



**Figure 4.2** Average percentage of number of macroblocks coded in each mode. The numbers are rounded for the clarity of the graph.

of the macroblocks. The prevalence of SKIP and INTER modes over INTRA modes is due to the high temporal correlation between subsequent video frames.

When Fig. 4.2 is analyzed in conjunction with Fig. 4.1, it is observed that the negligible complexity of the SKIP mode trial is in striking contrast to its occurrence. For the 35 percent of the macroblocks that are eventually coded in SKIP mode, the RDO mode decision has to evaluate the remaining INTRA and INTER modes to conclude that the SKIP mode is in fact the optimal mode. Our conclusion from this observation is that the encoder complexity could be significantly reduced if the SKIP coded macroblocks or, even better, the approximate final RD costs of the macroblocks were known prior to mode decision. The potential benefits of early detection of SKIP coded macroblocks have also been noted by other researches in the past [19] and [18]. In the pursuit of this inference, we have designed a novel RD cost prediction method which predicts the RD cost of a macroblock by exploiting the readily available information from the previous frame prior to any mode trials.

## 4.2 RD-cost Prediction<sup>1</sup>

Inspired from the work of Kannangara et al. [19], we hypothesize that the distortion and rate values of co-located macroblocks in temporally neighboring frames are correlated such that the RD cost of a macroblock in the current frame can be accurately estimated using the distortion and rate values of the co-located macroblock in the previous frame. Following this hypothesis, for a macroblock  $X_i^n$  in frame  $n$ , we use the distortion,  $D_i^{n-1}$  and rate,  $R_i^{n-1}$  of the co-located macroblock  $X_i^{n-1}$  in frame  $n - 1$  to predict its RD cost as follows:

$$\hat{J}_i^n = \alpha_d \cdot D_i^{n-1} + \alpha_r \cdot \lambda \cdot R_i^{n-1} \quad (4.1)$$

The distortion and rate values are scaled by factors  $\alpha_d$  and  $\alpha_r$  respectively. The optimal  $\alpha$  pairs are obtained empirically through simulations ran for all 15 sequences with QP varied between 8 and 48.  $\alpha_d$  and  $\alpha_r$  are incremented in steps of 0.1 starting from 0.1 up to 2.0. For each sequence-QP combination, the empirical optimal  $\alpha$  pairs are recorded as the pairs minimizing the average squared error between the actual and the predicted RD costs. Kannangara et al. assume fixed values for  $\alpha_d$  and  $\alpha_r$ , 1.0 and 0.5 respectively in their early SKIP termination work [19]. However, simulations show that both  $\alpha_d$  and  $\alpha_r$  depend on the quantization parameter; therefore, we model the  $\alpha$ 's as functions of the quantization parameter. We tried fitting various degrees of functions to the experimental data but the second degree polynomials gave a reasonable approximation with negligible complexity overhead. The resulting  $\alpha$  functions are as follows:

$$\alpha_d = -0.0001 \cdot QP^2 + 0.0069 \cdot QP + 0.9809 \quad (4.2)$$

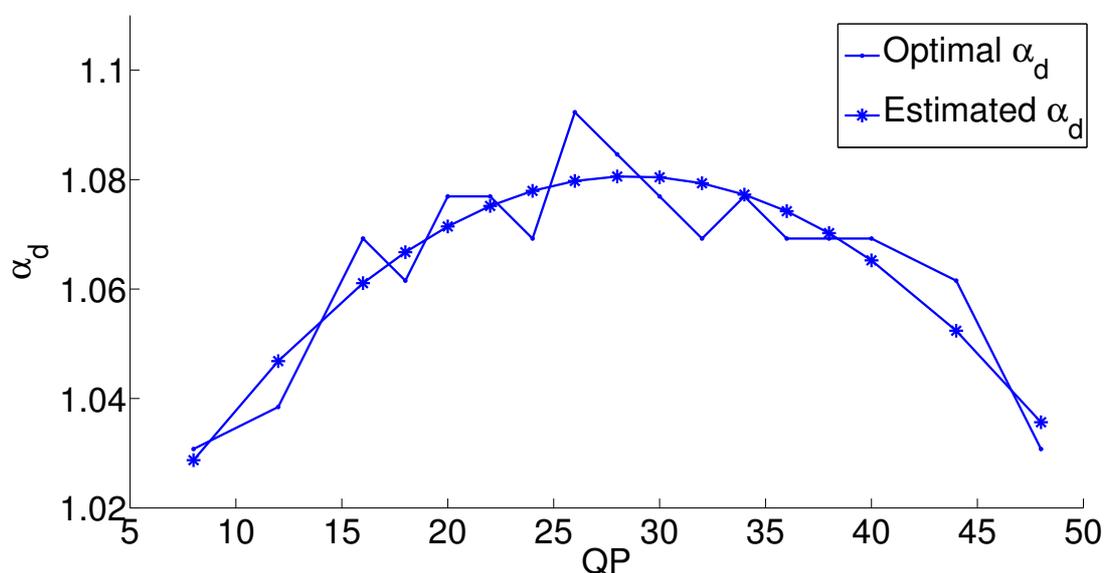
$$\alpha_r = -0.0002 \cdot QP^2 - 0.0025 \cdot QP + 1.0164 \quad (4.3)$$

The figures 4.3 and 4.4 plot the optimal  $\alpha_d$  and  $\alpha_r$  values along with their second degree polynomial estimations against different quantization parameters. The figures clearly illustrate that the actual  $\alpha$  values are accurately approximated by the second degree polynomials. It is important to note that while the mean of  $\alpha_d$  is 1.06, the mean of  $\alpha_r$  is 0.73. The difference in the means is due to the fact that we are trying to represent the macroblock similarity by estimating the RD costs. Therefore, when the co-located MBs

---

<sup>1</sup>This section of the work will be presented at the International Green Computing Conference [26].

are similar, it is possible to predict one from the other accurately using either the SKIP mode or an INTER mode with large partitions and small motion vectors. This eventually decreases the rate of a coded macroblock; thus the decline in the  $\alpha_r$ . Additionally,  $\alpha_r$  decreases with increasing QP because an increase in QP decreases the detail level of the images and increases the MB similarity in different frames. Once again, this increases the number of MBs that are skipped or coded with large partitions and small motion vectors; thus decreasing the bit-rate and the  $\alpha_r$  values.



**Figure 4.3** Optimal  $\alpha_d$  and its second degree polynomial estimation.

Fig. 4.5 displays the histogram graph for the absolute values of the macroblock RD cost prediction error percentages,  $\left| \frac{J_i^n - J_i^n}{J_i^n} \times 100 \right|$ , over all the simulated sequences and QPs. The estimated  $\alpha$  values are used in all the simulations from this point. We see that the prediction error is mostly limited to forty percent. The mean and the variance of the prediction error are 10.67 and 155.70 respectively.

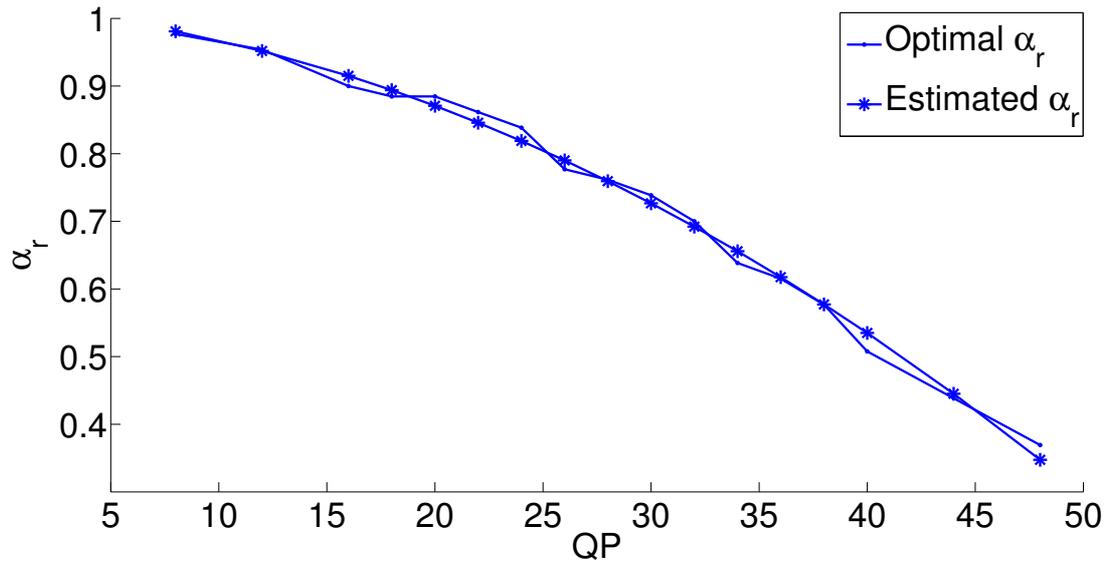
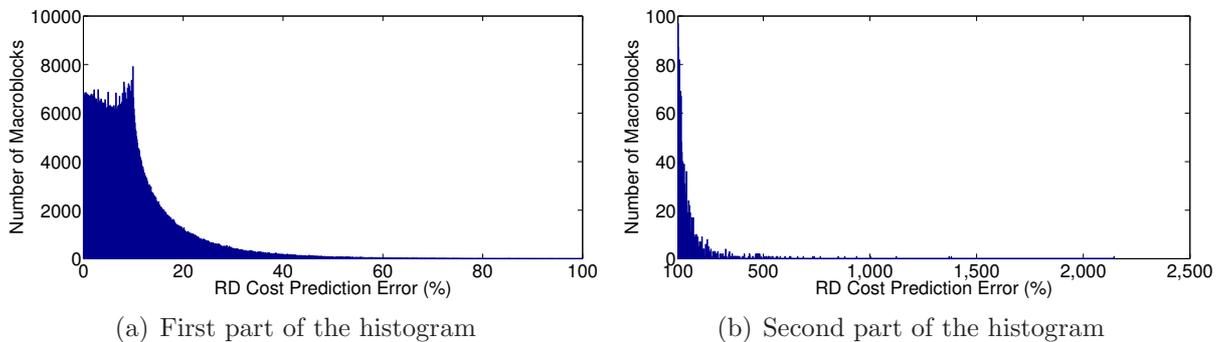


Figure 4.4 Optimal  $\alpha_r$  and its second degree polynomial estimation.



(a) First part of the histogram

(b) Second part of the histogram

Figure 4.5 Histogram of the absolute macroblock RD cost prediction error percentages for 15 different sequences and 17 QPs.

### 4.3 Application of RD-cost Prediction on Early Skip Termination<sup>2</sup>

Using equation (4.1), the RD cost prediction method computes a prediction of the final RD cost of a macroblock prior to mode decision. The significance of this method is that the predicted RD cost serves as a benchmark figure between different mode trials. In fact, we propose an Early Skip Termination (EST) algorithm which uses this predicted RD cost to decide whether a macroblock should be SKIP coded or not, without trying out any INTRA or INTER modes. For a macroblock, the EST algorithm first evaluates the SKIP mode and compares the predicted RD cost  $\hat{J}_i^n$  with the RD cost of SKIP  $J_i^n(SKIP)$ . If SKIP has a cost less than or equal to the predicted cost, then the mode decision process is terminated and the macroblock is skipped. A block diagram of the algorithm is illustrated in Fig. 4.6 and the steps are listed below:

1. For each macroblock, using equation (4.1) predict the RD cost prior to any mode trials.
2. Evaluate SKIP mode and obtain the corresponding RD cost.
3. If the RD cost of SKIP mode is less than or equal to the predicted RD cost, terminate the RDO mode decision process and code the macroblock with the SKIP mode.
4. If not, proceed with regular RDO mode decision steps and evaluate the INTRA and INTER modes.

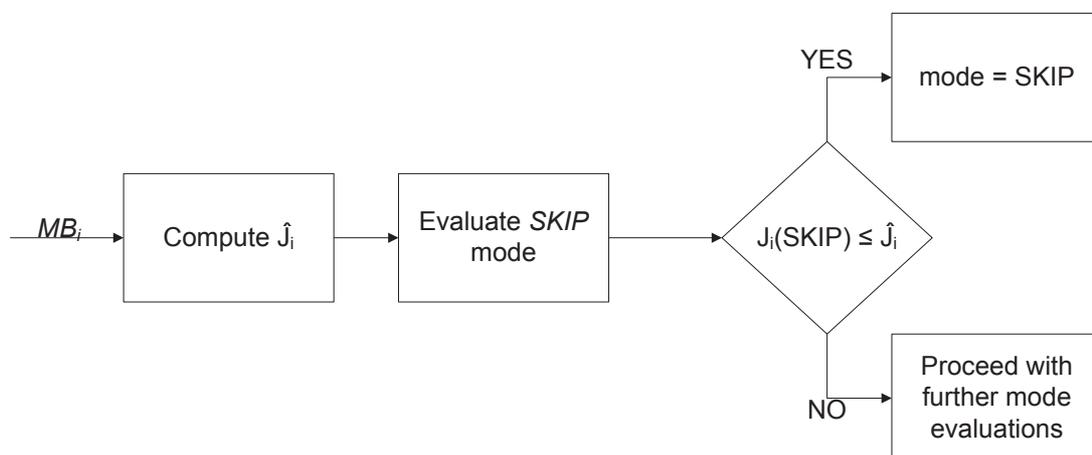
The basic premise of EST is to identify the macroblocks that are likely to be skipped prior to any INTRA and INTER mode trials; thus relieving the encoder of the computational burden of evaluating INTRA and INTER modes for some macroblocks.

#### 4.3.1 Experimental Results

In order to validate the performance of the proposed Early Skip Termination method, we follow two measures of comparison. First, the encoding time savings are compared with the Early Skip Termination method proposed by Jeon and Lee [18] that is included in

---

<sup>2</sup>This section of the work will be presented at the International Green Computing Conference [26].



**Figure 4.6** Block diagram of the Early SKIP Termination algorithm

the reference JM software. Secondly, the rate-distortion performance of EST is compared with the rate-distortion-optimized mode decision. The results have been tested for all 15 sequences and all perform similarly; yet four sequences are chosen to demonstrate the findings. These four sequences are Akiyo, Silent, Foreman and Football and they cover the necessary variety of motion content to illustrate the proposed algorithm’s performance.

Table 4.1 records the percentages of the encoding time saved by the proposed EST and by the early SKIP termination method implemented in the JM reference software. EST reduces the encoding time by over 80 percent in some cases notably surpassing the early SKIP termination method used in JM reference software. Table 4.2 presents three different sets of data for different QPs. The first row, namely the “Correct SKIP MBs”, contains the percentages of macroblocks that are skipped by EST and by the RDO mode decision with respect to the total number of macroblocks that are skipped by the RDO mode decision. The second row, namely the “Missed SKIP MB”, contains the percentages of macroblocks that are not skipped by EST but skipped by the RDO mode decision with respect to the total number of macroblocks that are skipped by the RDO mode decision. Finally, the third row, namely the “New SKIP MBs”, contains the percentages of macroblocks that are skipped by EST but not skipped by the RDO mode decision with respect to the total number of macroblocks that are skipped by the RDO mode decision. These are the average percentages for the 15 sequences. The RD impact of the “missed” and the “new” SKIP MBs are insignificant as the rate-distortion performances of EST and RDO encoders for

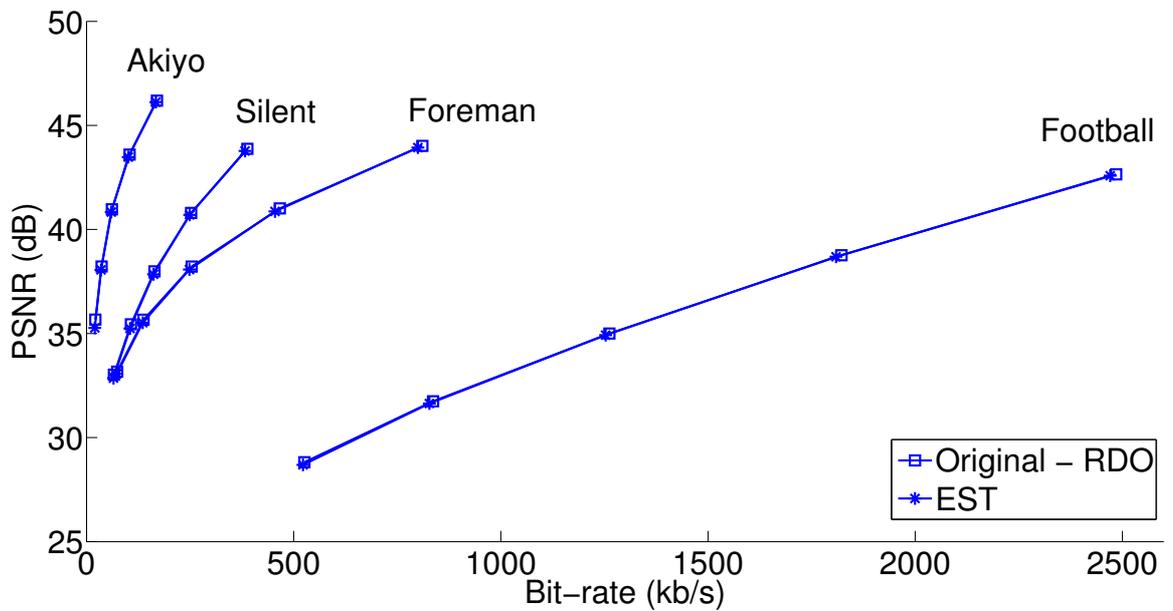


Figure 4.7 RD curves of the proposed EST method

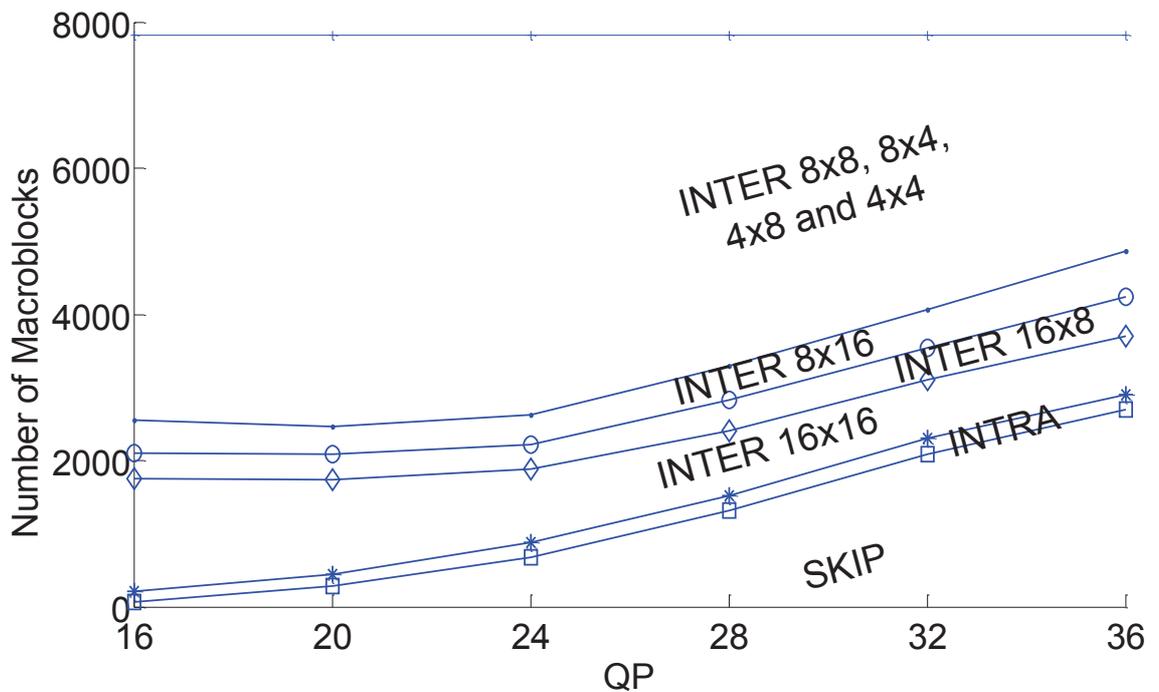


Figure 4.8 RDO Mode decision representation by distance-between-the-curves method as QP varies for Football sequence

the four sequences are almost identical as shown in Fig. 4.7. It is clearly seen that the offsets between two sets of RD curves are indiscernible.

**Table 4.1** Comparison of The Saved Encoding Time Percentages by The Proposed EST and by The EST Adopted in JM Reference Software. The Total Encoding Time for JSVM and JM encoders in RDO mode is also presented.

Sequence	Encoder	QP16	QP20	QP24	QP28	QP32
Akiyo	JSVM RDO (Enc. time in sec.)	88.72	88.30	88.09	87.86	88.10
	EST (Enc. time savings in %)	<b>65.42</b>	<b>69.67</b>	<b>74.39</b>	<b>78.57</b>	<b>82.62</b>
	JM RDO (Enc. time in sec.)	27.23	26.15	24.02	22.63	22.18
	JM EST (Enc. time savings %)	29.76	45.34	33.78	39.29	40.61
Silent	JSVM RDO (Enc. time in sec.)	88.73	88.39	88.23	88.04	88.13
	EST (Enc. time savings in %)	<b>45.24</b>	<b>52.50</b>	<b>57.59</b>	<b>62.60</b>	<b>66.88</b>
	JM RDO (Enc. time in sec.)	34.14	32.48	30.56	29.22	27.95
	JM EST (Enc. time savings in %)	8.50	22.77	33.78	39.29	40.61
Foreman	JSVM RDO (Enc. time in sec.)	89.11	88.71	88.47	88.49	88.21
	EST (Enc. time savings in %)	<b>10.76</b>	<b>19.44</b>	<b>27.59</b>	<b>33.99</b>	<b>40.74</b>
	JM RDO (Enc. time in sec.)	46.83	42.75	38.01	33.95	31.13
	JM EST (Enc. time savings in %)	-0.12	1.91	3.51	9.22	15.58
Football	JSVM RDO (Enc. time in sec.)	89.03	88.41	88.72	88.53	88.33
	EST (Enc. time savings in %)	<b>2.99</b>	<b>5.71</b>	<b>10.60</b>	<b>17.73</b>	<b>26.26</b>
	JM RDO (Enc. time in sec.)	68.74	65.82	62.63	59.52	56.58
	JM EST (Enc. time savings in %)	-0.02	-0.13	0.30	1.22	4.67

It should be noted that both Early SKIP Termination algorithms achieve higher complexity savings with slow motion sequences and with high quantization parameters. This is due to the fact that, in slow motion sequences, frames are more temporally correlated and more macroblocks are skipped which enables the benefits of the Early SKIP Termination methods to be exercised in a larger fraction of macroblocks. Similarly, higher quantiza-

tion parameters decrease the spatial details in frames; hence the variance of the difference between the pixel values of temporally consecutive frames decreases. When temporally frames are highly correlated, the encoder can use larger block sizes to approximate the motion between the frames hence the INTER mode  $P16 \times 16$  and especially the SKIP mode are used to code a greater percentage of the macroblocks. This phenomenon is illustrated in Fig. 4.8 as the number of macroblock mode decisions are graphed in cumulative manner. The distance between each curve signifies the number of macroblocks coded in a mode as pointed out on the figure. In order to generate this graph, the Football test sequence was encoded without EST.

**Table 4.2** Comparison of The Relative Percentages of The Skipped MBs in EST and in RDO Mode Decision

	QP16	QP20	QP24	QP28	QP32	QP36	QP40
Correct SKIP MBs	89.48	89.74	89.55	88.21	88.66	90.21	90.85
Missed SKIP MBs	10.52	10.26	10.45	11.79	11.34	9.70	9.15
New SKIP MBs	19.57	21.63	21.32	21.22	19.9	16.71	14.25

In view of Table 4.1 and Fig. 4.7, two conclusions are drawn. First is that the EST can successfully identify the MBs that are likely to be skipped. The second is that the RD-cost prediction method introduced in Section 4.2 can accurately estimate the RD cost of a macroblock.

#### 4.4 Application of RD-cost Prediction on Early Mode Termination

The performance analysis of Early SKIP Termination on different sequences highlighted several critical points. While EST achieves significant complexity savings for slower sequences, it falls short of achieving the same degree of savings for faster sequences as the number of SKIP coded macroblocks in such sequences is insufficient for EST to perform at a desired level. On the other hand, the negligible impact on rate-distortion performance reassures the accuracy of the proposed RD cost prediction method.

Motivated by providing higher computational savings for faster sequences and by the attested RD cost prediction, we extended the early termination criterion of EST to the remainder of the modes. This extension forms the basis of new complexity reduction tool called the Early Mode Termination (EMT) algorithm. Early Mode Termination codes a macroblock with the first mode that has an RD cost less than or equal to its predicted RD cost.

In EST, given that the first evaluated mode is SKIP, the order in which the rest of the modes are evaluated does not affect the encoder complexity. However in EMT since the early termination criterion is also applied to the INTER and INTRA modes, the mode evaluation order in its entirety affects the encoder complexity and should be carefully chosen. Our goal was to determine a single order with ascending computational complexity that could be used for any sequence. At this point, referring back to Fig. 4.1, we reevaluate the mode decision complexity distribution and derive a mode mapping function,  $ModeToTest()$ , as presented in Table 4.3. This function dictates the next mode to be evaluated based on the previously evaluated mode and follows an ascending computational complexity.

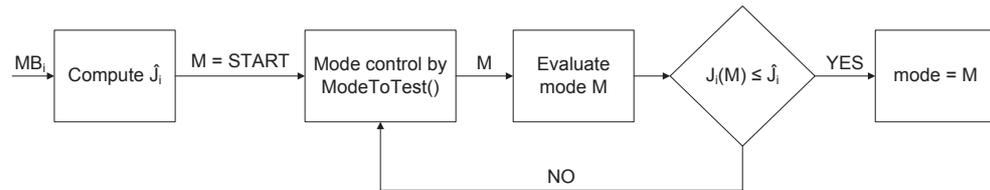
**Table 4.3** The Definition of The Mode Mapping Function  $ModeToTest$

$ModeToTest(.)$							
Input	$START$	Skip	Intra	$P16 \times 16$	$P16 \times 8$	$P8 \times 16$	$P \leq 8 \times 8$
Output	Skip	Intra	$P16 \times 16$	$P16 \times 8$	$P8 \times 16$	$P \leq 8 \times 8$	N/A

A block diagram of the Early Mode Termination mode decision is illustrated in Fig. 4.9 and the steps are as follows:

1. For each macroblock, using equation (4.1) predict the RD cost prior to any RDO mode decision step.
2. Set mode  $M$  to  $START$ .
3. Set mode  $M$  to  $ModeToTest(M)$  and evaluate mode  $M$ .
4. If the RD cost of mode  $M$  is less than or equal to the predicted RD cost, terminate the RDO mode decision process and code the macroblock in mode  $M$ .

5. If not, go to step 3.



**Figure 4.9** Block diagram of the Early Mode Termination algorithm

#### 4.4.1 Experimental Results

We tested the Early Mode Termination algorithm with all 15 sequences and obtained favorable results for each one. However, since the motivation behind developing EMT was to provide a better complexity reduction tool for faster sequences, we chose four of the faster sequences to present the performance of the algorithm: Foreman, Stefan, Football and NBA.

Similar to the analysis of the Early SKIP Termination algorithm, we will follow two measures of comparison for EMT as well. We will compare the percentages of EMT's encoding time savings with EST's. However, the rate-distortion performance will also be compared with the rate-distortion-optimized mode decision as the RDO mode decision sets the optimal performance.

Table 4.4 compares the percentages of the saved encoding time by EMT and by EST. Clearly EMT achieves higher complexity savings. In fact, the improvement over EST is more evident for smaller QPs where EST barely saves any encoding time and EMT saves at least 25 percent of the encoding time. Rate-distortion performance of EMT is compared with EST and RDO mode decision for the four sequences as shown in Fig. 4.10. Although the difference between the RD curves of EMT and RDO is slightly more discernible than the difference between EST and RDO, it is still negligible.

## 4.5 Summary

The chapter opened with the complexity analysis of the RDO mode decision process. The complexity breakdown of mode decision was investigated through profiling the execution of

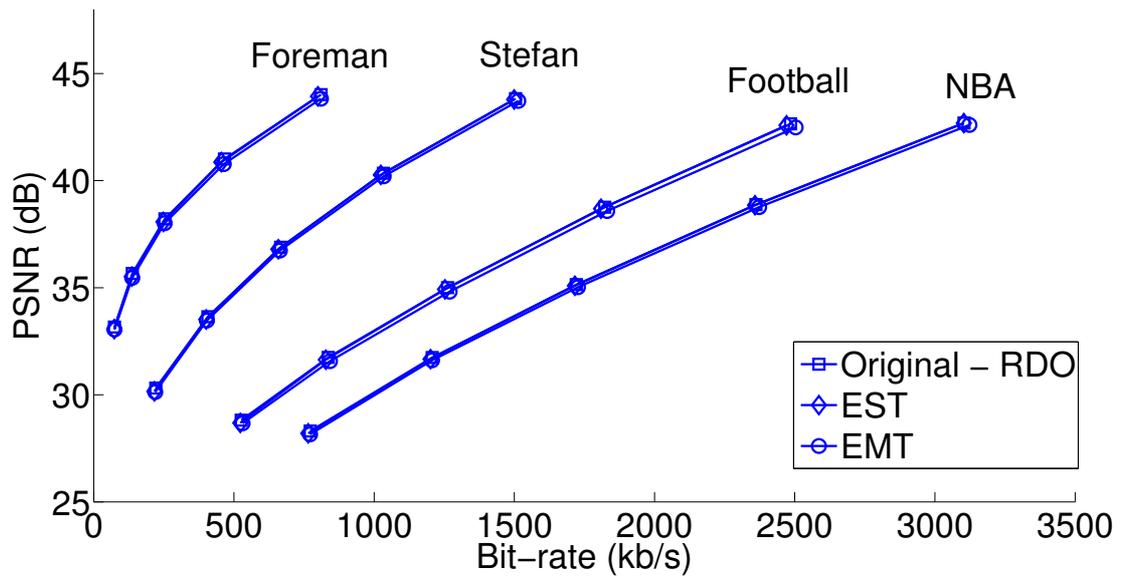


Figure 4.10 RD curves of the proposed EMT method

Table 4.4 Comparison of The Saved Encoding Time by The Proposed EST and by The Proposed EMT

Sequence	Encoder	QP16	QP20	QP24	QP28	QP32
Foreman	EST	10.76	19.44	27.59	33.99	40.74
	EMT	<b>41.89</b>	<b>46.13</b>	<b>49.39</b>	<b>54.07</b>	<b>57.69</b>
Stefan	EST	5.95	9.64	14.36	21.05	30.64
	EMT	<b>34.72</b>	<b>36.64</b>	<b>38.69</b>	<b>41.88</b>	<b>48.40</b>
Football	EST	2.99	5.71	10.60	17.73	26.26
	EMT	<b>33.25</b>	<b>35.00</b>	<b>38.47</b>	<b>43.63</b>	<b>52.67</b>
NBA	EST	0.03	1.00	2.41	4.32	7.99
	EMT	<b>27.76</b>	<b>25.93</b>	<b>24.99</b>	<b>24.39</b>	<b>27.25</b>

the encoder. Additionally, the relation between the number of macroblocks coded with the SKIP mode and the complexity of the SKIP was discussed. In fact, this discussion led to the introduction of a new RD cost prediction algorithm to be used in the reduction the encoder complexity. The prediction algorithm uses the co-located macroblock in the previous frame to estimate the RD cost of a macroblock in the current frame. The performance of the prediction was confirmed with a set of experimental results.

After establishing the RD cost prediction algorithm, a novel Early SKIP Termination (EST) algorithm was introduced. EST reduces the encoding time by using the predicted RD cost to predetermine the macroblocks that will be coded with the SKIP mode. Simulation results proved that EST surpasses the early SKIP termination algorithm implemented in the JM reference software [18]. However, the encoding time savings were found to be unsatisfactory for fast sequences. This led to the development of another complexity reduction algorithm, the Early Mode Termination (EMT) which is an extension of EST. EMT evaluated the modes in an ascending complexity order and uses the first mode with an RD cost less than or equal to the predicted RD cost to code the macroblock with. Simulation results showed substantial improvements in encoding time savings over EST particularly for faster sequences with still negligible loss in RD performance.

It is important to note that neither EST nor EMT provide scalability over the encoder complexity. They are merely complexity reduction tools that seek to save encoding time by identifying the coding modes of macroblocks in earlier stages of encoding. Although these tools may be used on their own, they will also be used as add-on's of the scalable encoding framework which will be explained in the next chapter. Hence, this concludes the presentation of the two complexity reduction tools as the next chapter will introduce a novel complexity scalable encoding framework.

## Chapter 5

# The Complexity Scalable Encoding Framework

This chapter introduces a new complexity scalable encoding framework. Section 5.1 opens the chapter by reiterating the necessity of complexity scalable frameworks for H.264 encoding and discusses some possible approaches to providing this scalability. Section 5.2 describes the scalable structure of the proposed framework in full detail. Section 5.3 illustrates how the RD cost prediction is used for a better resource allocation that improves the RD performance of the encoder working with reduced complexity. Section 5.4 presents the applications of the two new complexity reduction tools, EST and EMT, on the complexity scalable encoding framework. Finally, Section 5.5 summarizes the highlights of the chapter. The supporting simulation results are provided for each of these algorithms within their relevant section following the theory.

### 5.1 Introduction

The H.264/AVC video coding standard is designed to target a very broad range of application scenarios that covers all forms of digital video from High Definition TV broadcasting to streaming over the Internet, wireless and mobile networks. While the new encoding standard is currently used in most of these scenarios, the high computational complexity entailed with its superior compression efficiency is a concerning problem particularly for battery supplied mobile devices. Over the years, researches have developed methods

to tailor the encoding complexity for these devices which usually have limited processing capacities and limited power supplies. Since the mode decision operation constitutes the majority of this complexity in the H.264/AVC encoding framework, these methods focus primarily on managing the computational complexity of the mode decision operation. They can be grouped into two classes based on their approaches.

Low complexity encoding methods attempt to reduce the encoding time and complexity with minimal impact on the RD performance of the output. These methods replace some of the encoding components with faster and complexity optimized designs and they generally utilize information from the previously encoded data to expedite the overall encoding process. In Chapter 4, we introduced two new low complexity mode decision algorithms which use the distortion and the rate values from the previously encoded frame to predict the final RD costs of the macroblocks in the current frame. The predicted RD costs are used to anticipate the coding modes of macroblocks without having to go through all of the mode decision steps; thus accelerating the total encoding process. However, devices with limited processing power require more than reduced complexity. These devices are often subject to volatile resource constraints and low complexity algorithms provide a reduced yet rigid complexity; thus falling short of optimizing the complexity as per these changing constraints. The second class of methods, the complexity scalable encoding methods, provide the desired flexibility in the encoder complexity to meet such constraints.

Complexity scalability is a means of regulating the amount of processing power used by the encoder according to the pressing constraints on the host platform or the encoder itself, while still producing a satisfactory outcome. A complexity scalable encoder is one that offers a variety of operational modes with the encoder complexity descending from maximum to minimum as the achieved quality also degrades gracefully. The idea behind a complexity scalable encoder is to translate the high level constraints (i.e. at a frame level or more usually at a sequence level) into a systematic allocation of the present processing power down to individual processing units in a video. Most complexity scalable frameworks prefer a frame level allocation of the resources where frames are dropped if their encoding time exceeds some threshold. Such implementations are popular as using a high level processing unit simplifies the translation of the constraints that are already defined at high level; however, frame dropping results in low and choppy perceptual quality that disturbs a viewer gravely.

An H.264/AVC encoder processes video frames in units of macroblocks (MBs); therefore,

implementing a complexity scalability on a MB level would be an ideal solution. However, macroblocks within a frame bear dependencies to some of their neighbors due to the fact that INTRA coding relies on the neighbors of a macroblock to form a prediction. Careful consideration must be shown to adhere to these dependencies if a MB level scalable encoding framework is to be implemented.

In this chapter, we propose a singly-parameterized complexity scalable H.264/AVC compliant encoding framework. We adopt the “wave-front MB scheduling” technique that provides the level of flexibility required to maintain a judicious allocation of available resources at a macroblocks level. While the main architecture of the proposed encoder is inspired from the recent complexity scalability work of Tan, Lee, Tham and Rahardja [23], we have improved the complexity modeling and the resource allocation of their work as simulation results will prove. The next section will establish the foundation of the proposed encoder.

## 5.2 The Complexity Scalable Encoding Framework<sup>1</sup>

The proposed encoder utilizes a macroblock level scalability and distributes the available processing power over different macroblocks. However due to the interdependence of the neighboring macroblocks (every MB requires some of its neighbors to be encoded before itself, see INTRA prediction in Chapter 2), there are limitations to the level of scalability attainable at the MB level. The wave-front (WF) macroblock scheduling technique adopted in our framework provides a way of processing macroblocks which achieves this scalability without disturbing the interdependencies. Macroblocks in a frame are grouped together in wave-fronts such that the MBs in a wave-front are independent of each other and can be encoded simultaneously given that the previous wave-fronts are fully encoded. Fig. 5.1 illustrates the wave-front grouping of macroblocks. Each box symbolizes a macroblock and the numbers inside the boxes indicate the wave-fronts the MBs belong to.

The RDO mode decision encoder processes each frame in units of macroblocks. For each macroblock, it tries successively the SKIP, INTRA and INTER modes. In other words, the coding modes are iterated for each macroblock. The grouping of independent macroblocks in wave-fronts allows us to modify this rigid operation of RDO mode decision such that instead of iterating the modes for each macroblock, we can iterate the macroblocks for

---

<sup>1</sup>This section of the work will be presented at the International Green Computing Conference [26].

1	2	3	4	5	6	7	8	9	10	11
3	4	5	6	7	8	9	10	11	12	13
5	6	7	8	9	10	11	12	13	14	15
7	8	9	10	11	12	13	14	15	16	17
9	10	11	12	13	14	15	16	17	18	19
11	12	13	14	15	16	17	18	19	20	21
13	14	15	16	17	18	19	20	21	22	23
15	16	17	18	19	20	21	22	23	24	25
17	18	19	20	21	22	23	24	25	26	27

**Figure 5.1** MBs in wave-front #11 are highlighted.

each mode within a wave-front. While processing a wave-front (WF) for mode decision, the encoder has the ability and the flexibility to switch between the macroblocks in that WF at any point in time. Therefore, the new encoding framework processes each frame in units of wave-fronts and can distribute the available computational resources over these MBs. For each wave-front, it loops over the coding modes and for each coding mode, it loops over the macroblocks. This raises three questions:

1. How is the resource allocation done in this framework?
2. What is the order in which the coding modes are iterated for a wave-front?
3. What is the order in which the macroblocks are iterated for a coding mode?

The amount of available processing power is represented by the number of RD operations that the encoder can perform. An RD operation is defined in our setting as a mode trial. We have noted earlier in Chapter 4 that the complexities of different mode trials are also different. Referring back to Fig. 4.1, we observe that SKIP and INTRA modes are of much lower complexity than any INTER mode, and that within the seven INTER modes, the modes with smaller block sizes are of slightly larger complexity. Yet the differences

between the complexities of INTER modes is negligible when compared with the differences between the SKIP/INTRA modes and the INTER modes. According to this observation, we quantify the mode complexities by assigning each mode an appropriate number of RD operations as follows:

- The SKIP mode: 0 RD operations
- The INTRA modes: 0 RD operations
- Each of the seven INTER modes: 1 RD operations

Based on these numbers, a complete RDO mode decision operation on a macroblock requires seven RD operations. Therefore, we define an RD operation budget for a wave-front  $WF_i$  as follows:

$$N_{op}(WF_i, \beta) = 7 \times N_{MB}(WF_i) \times \beta, \quad (5.1)$$

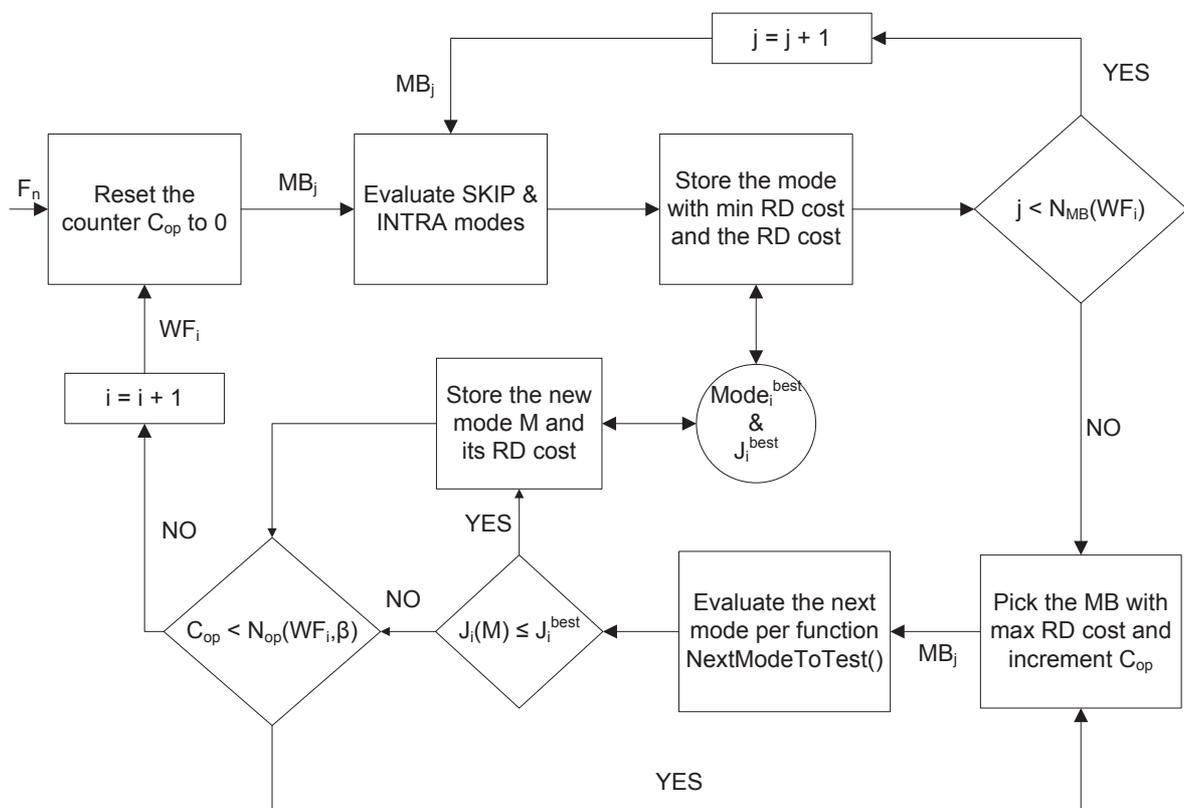
where  $N_{MB}(WF_i)$  is the number of macroblocks in wave-front  $WF_i$ .  $\beta$  is the single control parameter that sets the maximum number of RD operation allowed for a wave-front. It forms a link between the available processing power and the processing power made available to a wave-front by the encoder. It may be varied within the range of 0 to 1.0 per wave-front or per frame depending on the imposing resource constraint. Setting  $\beta$  to 1 amounts to allowing the encoder to explore all the mode trials within a wave-front.  $\beta$  is the sole parameter used to scale and control the encoder complexity.

**Table 5.1** The Definition of The Mode Mapping Function *NextModeToTest*

		<i>NextModeToTest(.)</i>						
Input	Skip/Intra	$P16 \times 16$	$P16 \times 8$	$P8 \times 16$	$P8 \times 8$	$P8 \times 4$	$P4 \times 8$	$P4 \times 4$
Output	$P16 \times 16$	$P16 \times 8$	$P8 \times 16$	$P8 \times 8$	$P8 \times 4$	$P4 \times 8$	$P4 \times 4$	N/A

So far, we have covered the allocation of the resources down to wave-fronts. The further distribution of these allocated resources over its macroblocks lies with the answers to the second and the third questions. As for the former, for each wave-front, the modes are iterated in an ascending complexity as dictated by the function *NextModeToTest* (see

Table 5.1). As for the latter, the available computational resources should be devoted to macroblocks with poorer coding performance. Since the RD cost of a macroblock is the indication of its coding performance, the next macroblock for which an RD operation will be spent, is the one with the maximum RD cost in the present wave-front. The motivation behind this practice is to give priority to MBs with worse RD performance in order to achieve a homogenous quality over a frame.



**Figure 5.2** The Block diagram of The Complexity Scalable Encoding of Frame  $F_n$ .

The complexity scalable encoding framework is completed and the block diagram of the encoder is shown in Fig. 5.2. The steps of the algorithm are as follows: for each wave-front,

1. Reset the counter to 0.
2. For each MB in the wave-front:
  - (a) Evaluate SKIP and INTRA modes.

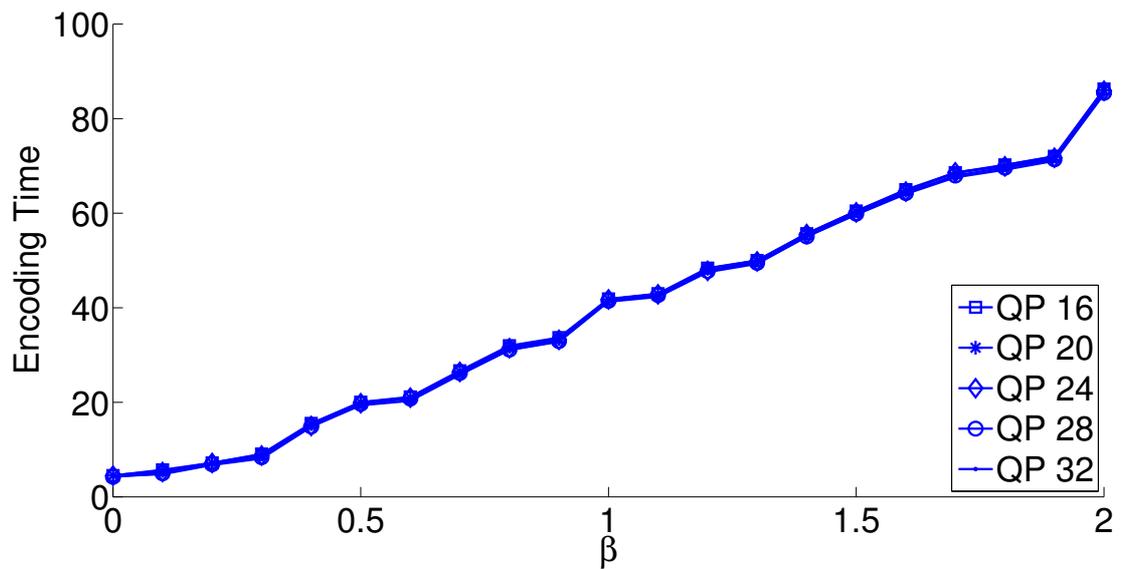
- (b) Store the mode with the minimum RD cost as the current best mode together with its RD cost.
3. Pick the MB with the maximum RD cost in the wave-front as the current MB.
4. Increase the counter of the number of operations by 1.
5. Evaluate the mode given by the output of the function *NextModeToTest* whose input is the current best mode of the MB in progress.
6. Update the current best mode with the newly tried one if the newly tried mode has a smaller RD cost.
7. Repeat steps from 3 to 7 until the counter is equal to  $N_{op}(WF_i, \beta)$ .

When compared with the complexity scalable framework proposed by Tan, Lee, Tham and Rahardja, our representation of each of the seven INTER modes as a single RD operation is a simple yet critical refinement over their consideration of the four INTER modes,  $P_8 \times 8$ ,  $P_8 \times 4$ ,  $P_4 \times 8$  and  $P_4 \times 4$  as a total of one operation [23]. As it will be demonstrated in Section 5.2.1, our assumption yields a more accurate modeling of the encoder complexity, in accordance with Fig. 4.1.

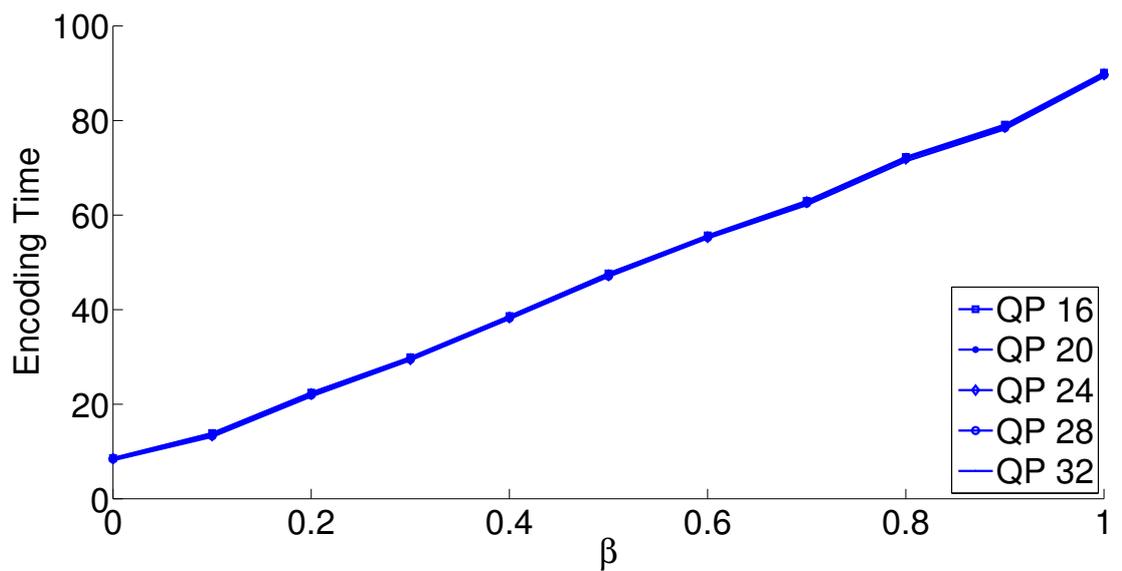
### 5.2.1 Experimental Results

In this section, the experimental findings on the proposed complexity scalable encoding framework (CSEF) are demonstrated. The scalability is maintained by varying the single control parameter  $\beta$  between 0 and 1.0 with increments 0.1.  $\beta$  determines the RD operation budget assigned for each wave-front; therefore it controls the complexity as well as the RD performance of the encoder. The simulation results will validate this postulate by plotting firstly the total encoding time and secondly the RD curves for different  $\beta$  values.

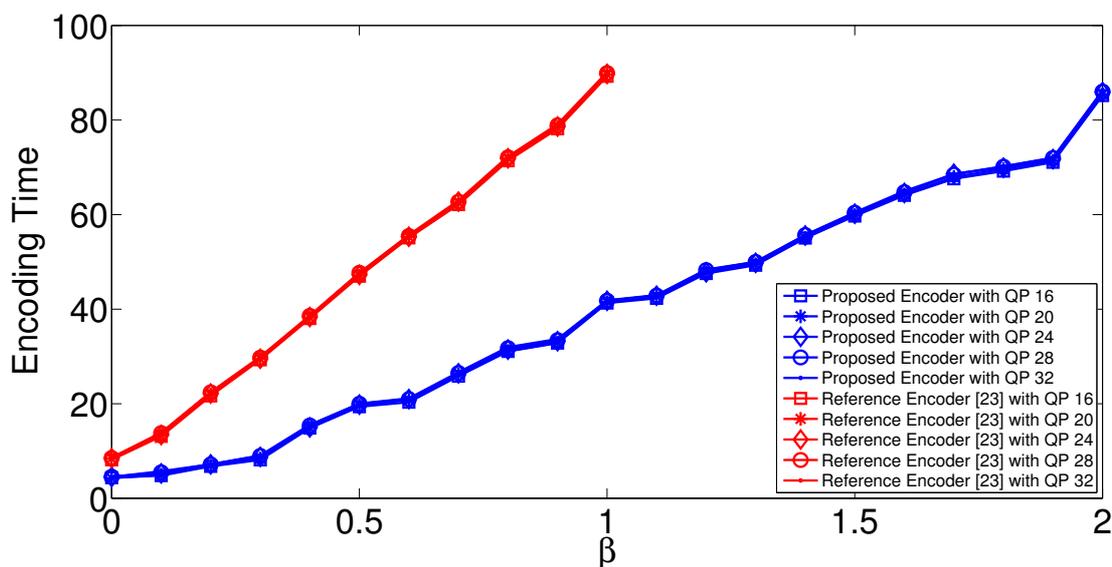
Initially, the encoder complexity modeling of our work will be compared with the modeling of the work of Tan, Lee, Tham and Rahardja [23]. The total encoding time recordings are plotted against  $\beta$  using different QPs for the two encoders in Figures 5.3 and 5.4, and compared with each other in Fig. 5.5. Although the plotted graphs are for the input test sequence Football, the encoding time pattern is constant over different sequences since there are not any complexity reduction tools employed that would modify the encoding steps



**Figure 5.3** Total encoding time in seconds vs.  $\beta$  curves for the complexity scalable framework proposed by Tan, Lee, Tham and Rahardja [23].



**Figure 5.4** Total encoding time in seconds vs.  $\beta$  curves for our proposed complexity scalable encoding framework.



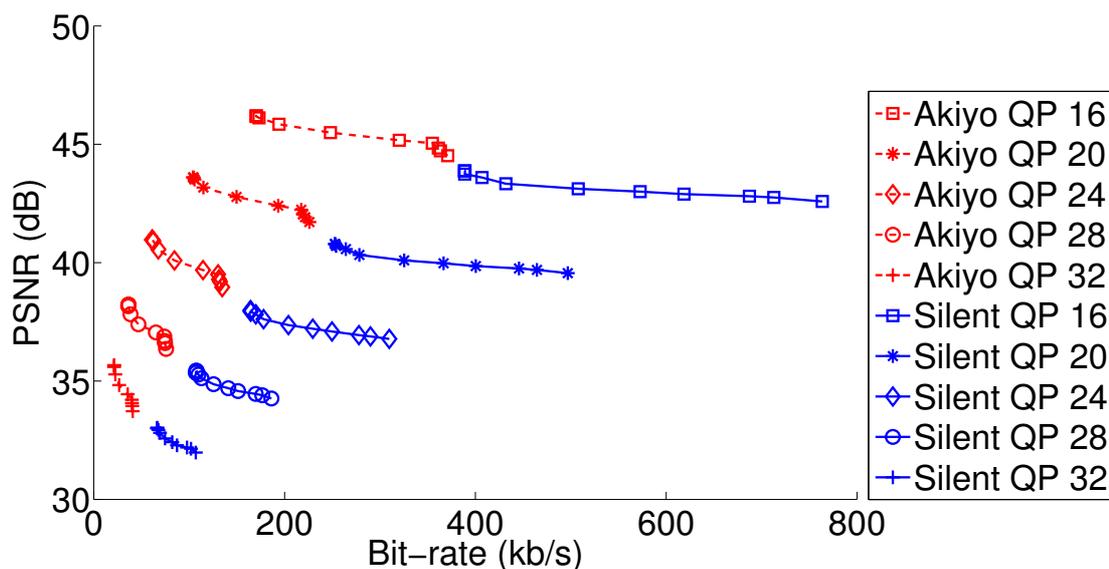
**Figure 5.5** Total encoding time in seconds vs.  $\beta$  curves compared for our proposed complexity scalable encoding framework and for the complexity scalable framework proposed by Tan, Lee, Tham and Rahardja [23].

depending on the characteristics of the input sequence (e.g. with EST, the encoding time of a slower sequence is shorter than of a faster sequence). Please also note the difference in the ranges of  $\beta$ . This is due to the fact that the proposed encoder performs a full RDO mode decision when  $\beta$  is equal to 1.0, whereas the reference encoder [23] performs a full RDO mode decision when  $\beta$  is equal to 2.0.

In a complexity scalable encoder, it is desirable to have a linear relation between the encoding time (i.e. the encoder complexity) and the complexity control parameter(s) of the encoder. Such a relation ensures that the effect of a complexity change on the encoding performance is predictable and maintainable. A linear relation signifies the preciseness of the complexity control of the encoder and the fine controllability of the complexity modeling. Clearly, the relation between the encoding time and the complexity control parameter  $\beta$  in Fig. 5.4 follows a more linear pattern than in Fig. 5.3. Hence, as hypothesized, the complexity modeling of our encoder is superior.

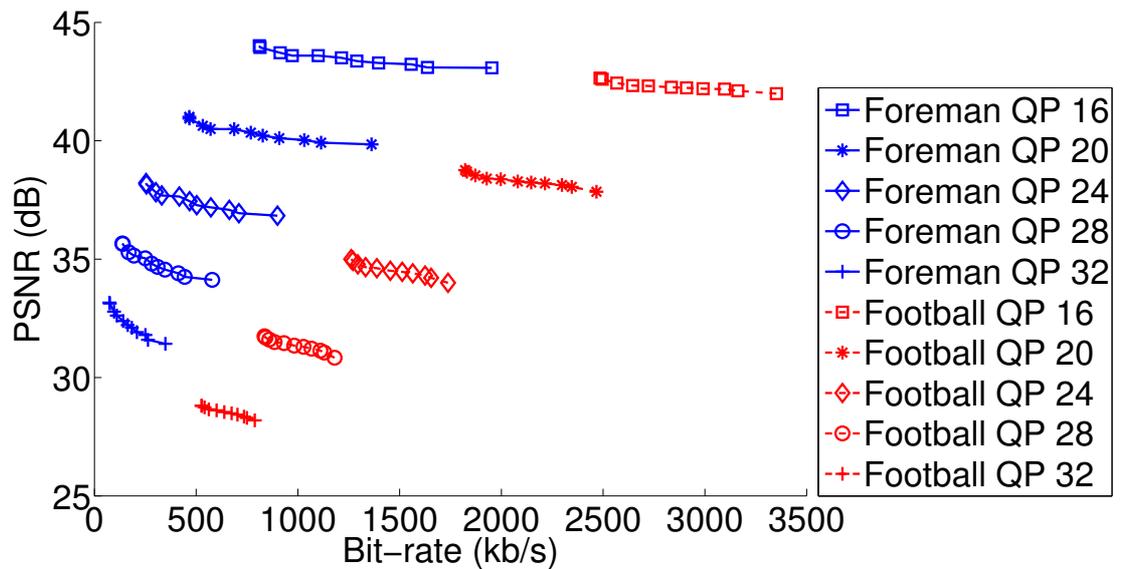
It is also imperative that the control parameter gradually scales the encoding quality in parallel with the complexity. Therefore, we expect the RD performance of our encoder to gradually improve as we increase  $\beta$  from 0.0 to 1.0 in steps of 0.1. Below, the RD curves for four sequences with five different QPs are presented. Fig 5.6 demonstrates the findings

for sequences Akiyo and Silent and Fig. 5.7 for sequences Foreman and Football. Each RD curve contains eleven different  $\beta$  points and as  $\beta$  increases, we observe a decreasing trend in bit-rate and an increasing trend in PSNR for all sequences and all QPs.

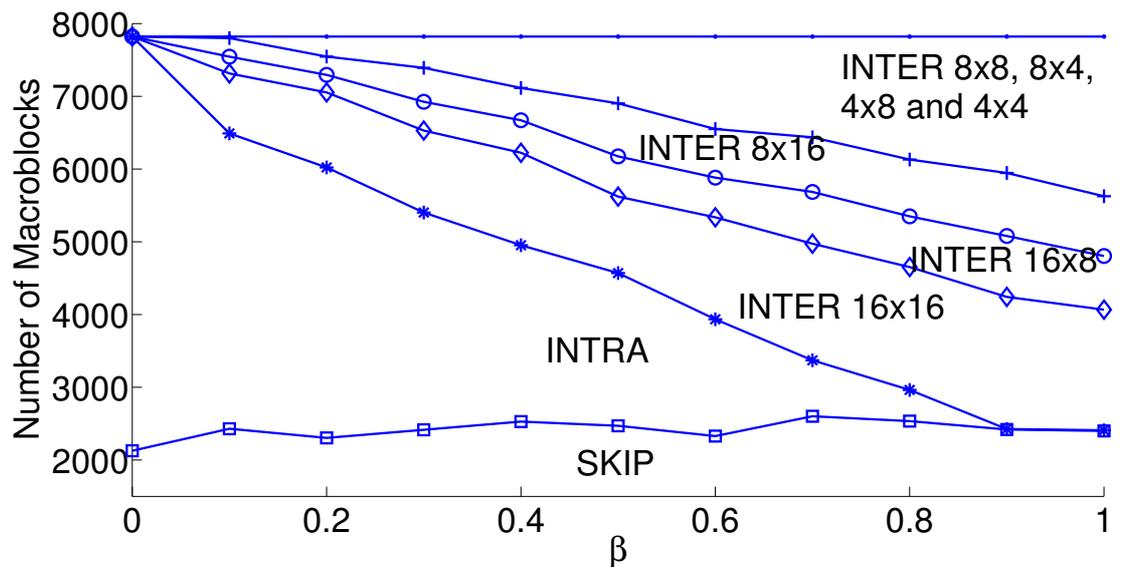


**Figure 5.6** RD curves for the proposed complexity scalable encoder for *Akiyo* and *Silent*, each curve is comprised of 11  $\beta$  values ranging from 0.0 to 1.0 by increments of 0.1.

The control parameter  $\beta$  can take eleven different values between 0.0 and 1.0 that are multiples of 0.1. For each value, the encoder operates at a different complexity and produces a useful outcome with an RD performance in accordance with its complexity. In other words, in these figures, the encoder has eleven operational modes with distinct computational complexity. The encoder complexity is set by restricting the number of operations, i.e. mode trials, performed per wave-fronts per Equation 5.1. When  $\beta$  is equal to 0.0, all the macroblocks are coded either with the SKIP or an INTRA mode. As it increases toward 1.0, INTER modes are evaluated more and more and when it is equal to 1.0, the complete RDO mode decision process is performed as for all the macroblocks, the encoder tries all the SKIP, INTRA and INTER modes. Fig. 5.8 illustrates the number of macroblocks coded in each mode as  $\beta$  varies for Foreman sequence with QP of 24.



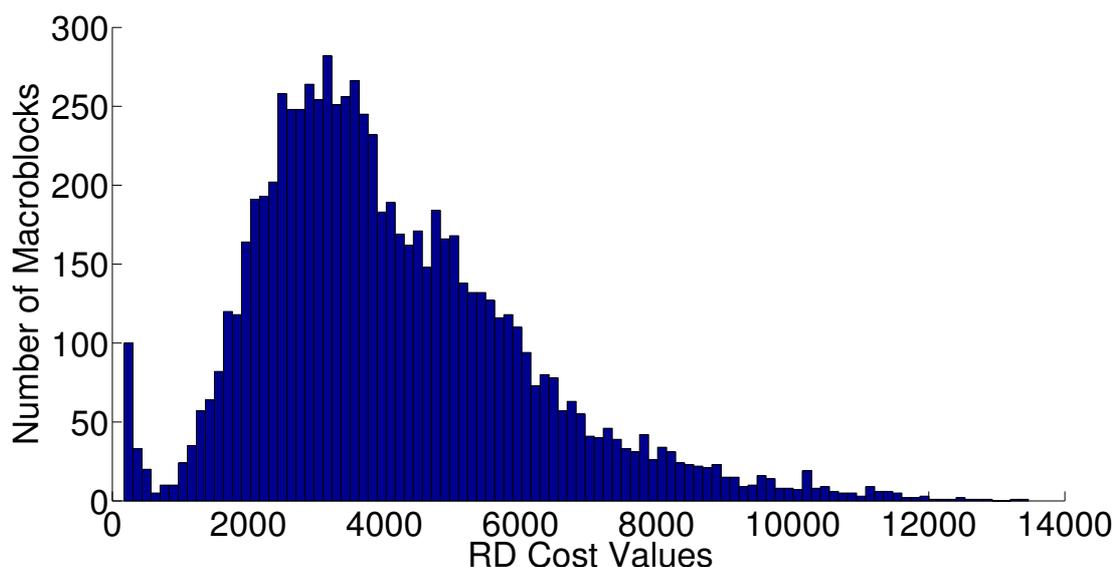
**Figure 5.7** RD curves for the proposed complexity scalable encoder for *Foreman* and *Football*, each curve is comprised of 11  $\beta$  values ranging from 0.0 to 1.0 by increments of 0.1.



**Figure 5.8** Mode decision representation by distance-between-the-curves method as  $\beta$  varies for *Foreman* with QP of 24.

### 5.3 Improved Resource Allocation: Application of RD-cost Prediction in Scalable Encoding

In Section 5.1, we said that a complexity scalable encoder should judiciously distribute the available computational resources over individual processing units. Consequently, we adopted the wave-front concept in which for each wave-front a budget of RD operations is allotted among its macroblocks. The allotment of each macroblock is based on its RD performance such that after each mode trial, the macroblock with the highest RD cost from the wave-front is granted another RD operation. This distribution method aims to achieve a uniform RD performance throughout the macroblocks in a frame by assuming that the RD costs of macroblocks are also uniform.



**Figure 5.9** Histogram of the macroblock RD cost values for Foreman sequence with a QP of 28.

Fig. 5.9 plots the histogram of the final RD costs of macroblocks for Foreman sequence encoded with a QP of 24 using full RDO mode decision. We observe a wide range of RD cost values extending from a little over 0 to as big as 14000. For instance, let us assume that there exists two macroblocks in a wave-front that is presently processed and that we know their final RD costs and their coding modes which are 1400 & SKIP and 100 & P\_4  $\times$  4. Let us also assume that our current complexity encoder allocated seven RD operation for this wave-front with two MBs and that so far it has tried SKIP and INTRA

modes for both MBs where the best RD costs are 1400 and 1000 respectively. Based on the algorithm steps defined in Section 5.2, the encoder will use all seven RD operations for the first MB with the RD cost of 1400 and ignore the second MB with the RD cost of 1000, not realizing that the first MB although with a larger RD cost has found its optimal RD cost and that the second MB with the smaller RD cost actually needs more mode trials and RD refinement. This is of course an extreme case constructed to illustrate a point; nevertheless it is plausible.

In search of smarter RD operation allocation design, we refer back to Section 4.2 where we introduced a novel RD cost prediction method which accurately estimates the RD cost of a macroblock using the information from the previous frame. Therefore, we incorporate the RD cost prediction in our complexity scalable encoder where the new steps to the encoding framework are now as follows: for each wave-front,

1. Reset the counter to zero.
2. For each MB in the wave-front:
  - (a) Compute the predicted RD cost  $\hat{J}_i$  by Equation 4.1.
  - (b) Evaluate SKIP and INTRA modes.
  - (c) Store the mode with the minimum RD cost and its RD cost as the current best mode  $\text{MODE}_i^{\text{best}}$  and as the current best RD cost  $J_i^{\text{best}}$ .
3. Pick the MB with  $\max(J_i^{\text{best}} - \hat{J}_i)$  as the current MB.
4. Increase the number of operations counter by 1.
5. Evaluate the mode given by the output of the function *NextModeToTest* whose input is the current best mode of the MB in progress.
6. Update the current best mode with the newly tried one if the newly tried mode has a smaller RD cost.
7. Repeat steps from 3 to 7 until the counter is equal to  $N_{op}(\text{WF}_i, \beta)$ .

The new step 3 of the algorithm gives priority to the macroblocks whose current RD cost has the maximum difference with its predicted RD cost. This practice takes into

consideration the fact that the RD costs of different macroblocks do not need to be same nor even similar and guarantees that a scenario similar to the exemplary one never occurs. The block diagram of the new encoding framework is presented in Fig. 5.10

### 5.3.1 Experimental Results

We expect the new resource allocation technique to improve the RD performance, particularly when  $\beta$  is small (less than 0.5). A small  $\beta$  value limits the allotted number of RD operations per wave-front and the new framework should do a smarter job of distributing the RD operations over the macroblocks. Therefore, we will compare the RD performances of the new complexity scalable framework with improved resource allocation with the original complexity scalable framework introduced in Section 5.2. However, we do not expect any change in the complexity modeling of the encoder and we will show that the relation between the encoding time and the complexity control parameter  $\beta$  is identical with the relation in the original framework.

Figures 5.11, 5.12, 5.13 and 5.14 plot the PSNR (in dB) and bit-rate (in %) gains of the new framework over the original framework for sequences Akiyo, Silent, Foreman and Football respectively. We observe considerable gains in both PSNR and bit-rate for all sequences and all QPs. In fact, as anticipated, the gain is especially significant for  $\beta$  values less than 0.5 because with such small values, the benefits of a smarter resource distribution are more noticeable. The encoding time for Football sequence is plotted against different  $\beta$  points for five different QPs in Fig. 5.15. The graph is almost identical to Fig. 5.4; thus the linearity of the complexity modeling is preserved. The corresponding RD curves for the encoding time graph is presented in Fig. 5.16. We can observe the incremental refinement in RD performance for increasing  $\beta$ .

We also would like to observe the impact of the new resource allocation technique on the final mode decisions. The final mode decisions for sequences Football and Foreman are plotted respectively in Figures 5.17 and 5.18 with and without the new resource allocation method. The mode decisions are the same for both encoders when  $\beta$  is equal to one. However, when  $\beta$  is less than one, we observe that the mode decision curves converge in a much smoother fashion with the new method.

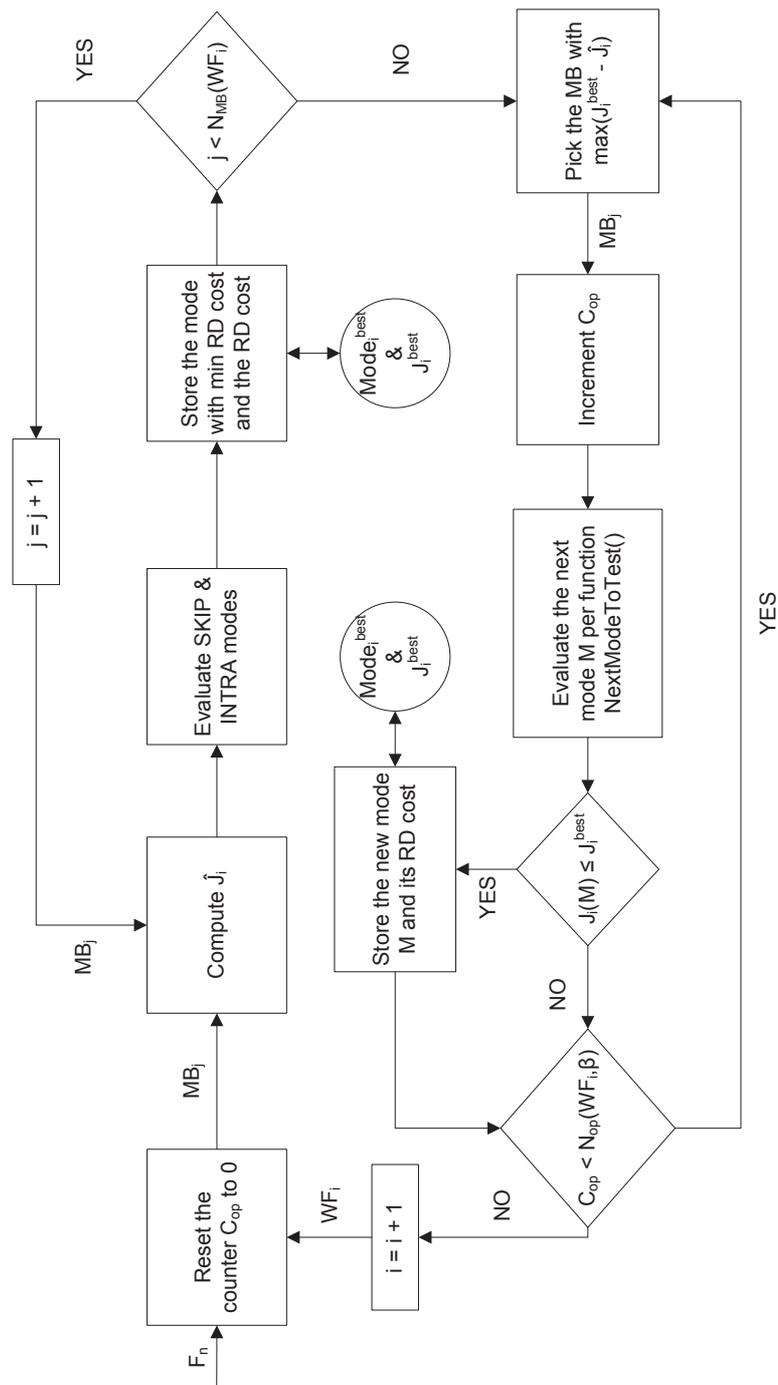
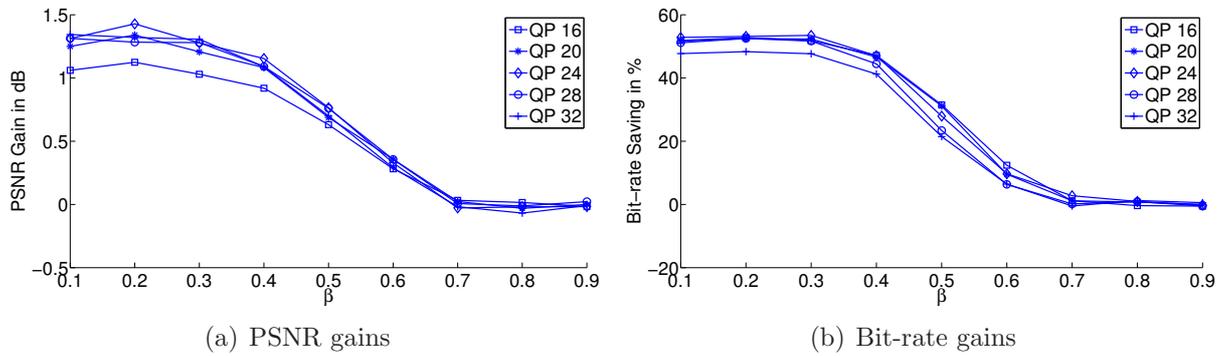
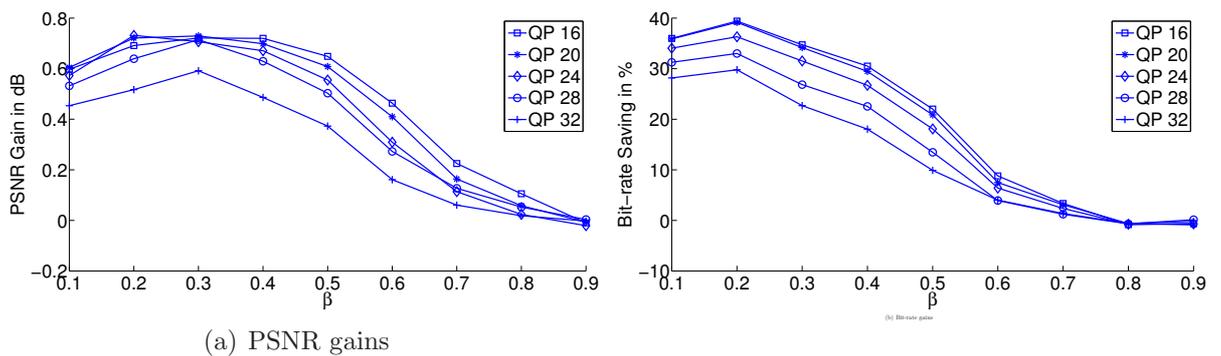


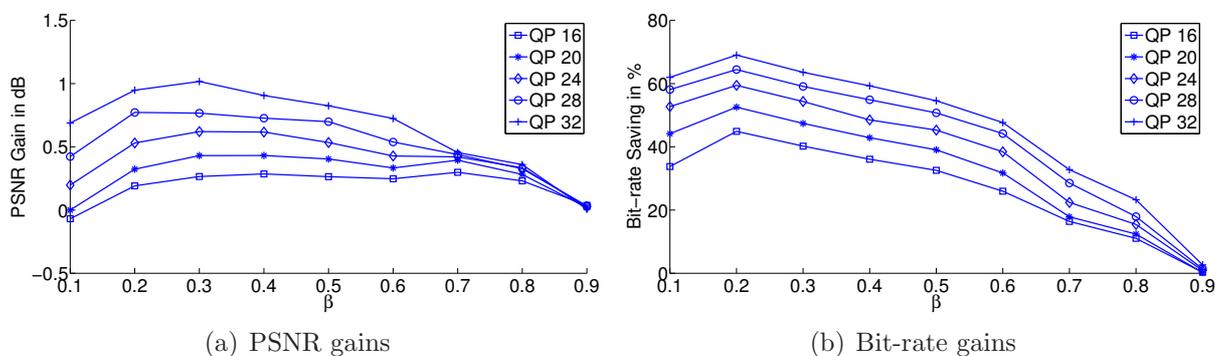
Figure 5.10 The Block diagram of The Complexity Scalable Encoding of Frame  $F_n$  with Improved Resource Allocation.



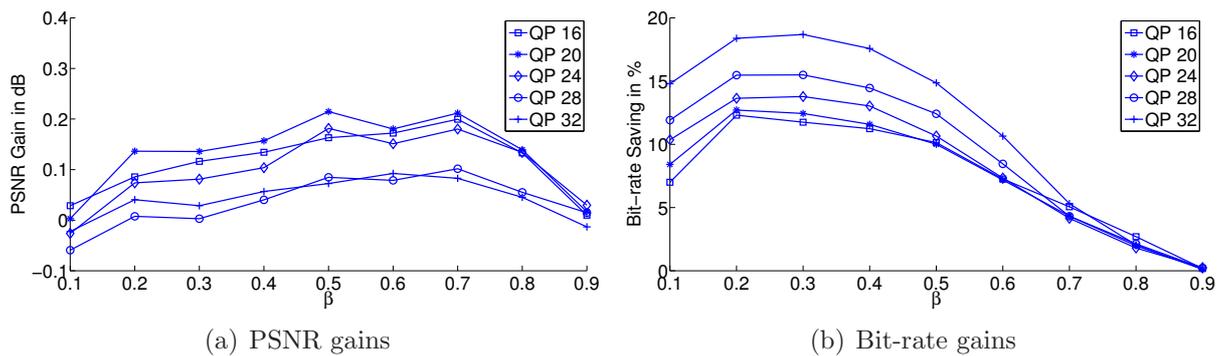
**Figure 5.11** PSNR and Bit-rate gains with improved resource allocation over different  $\beta$  values for Akiyo sequence



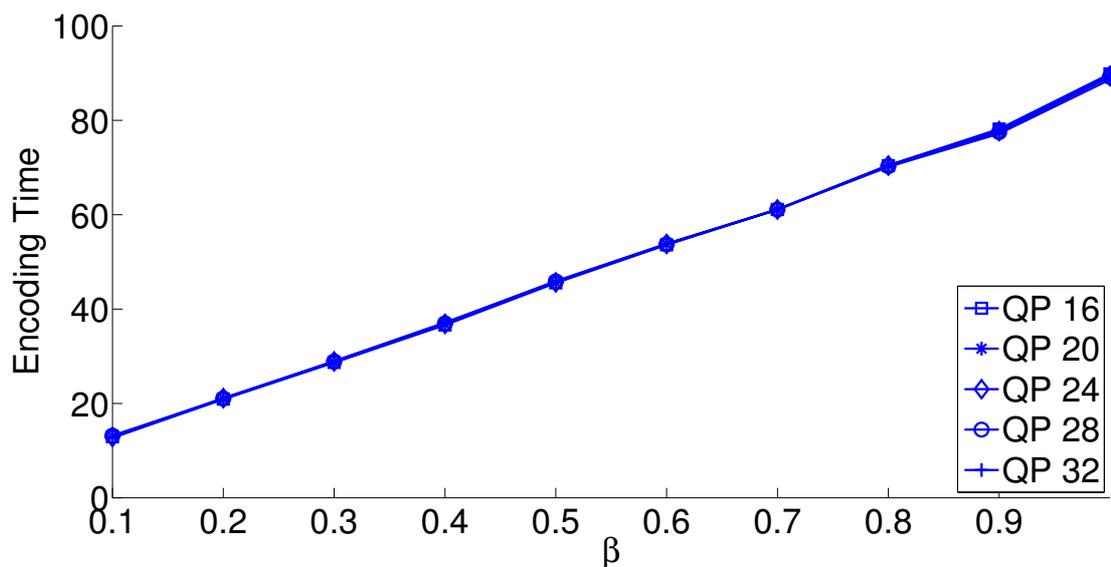
**Figure 5.12** PSNR and Bit-rate gains with improved resource allocation over different  $\beta$  values for Silent sequence



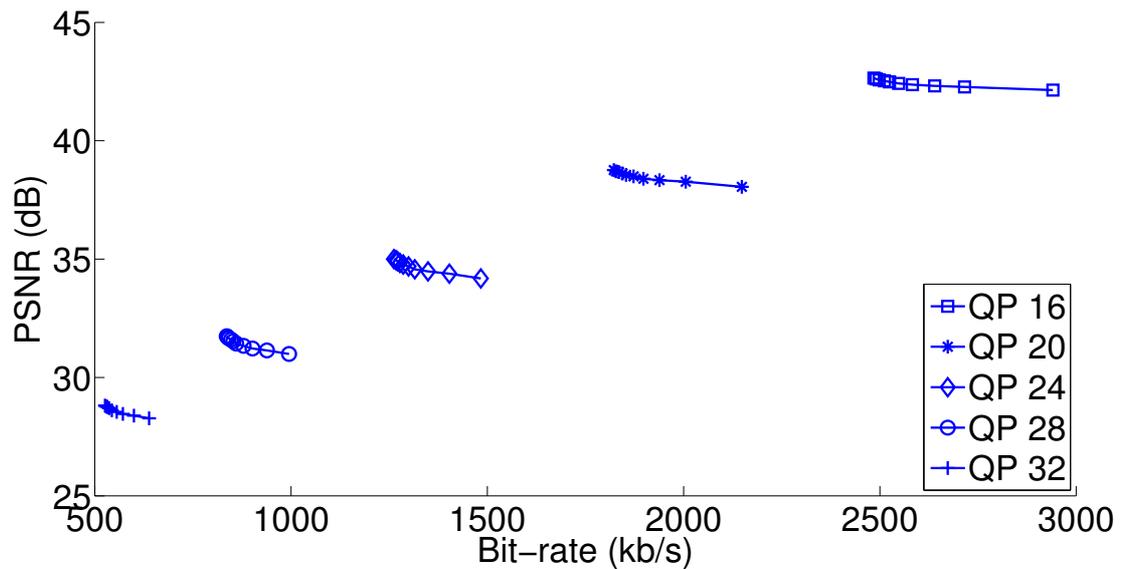
**Figure 5.13** PSNR and Bit-rate gains with improved resource allocation over different  $\beta$  values for Foreman sequence



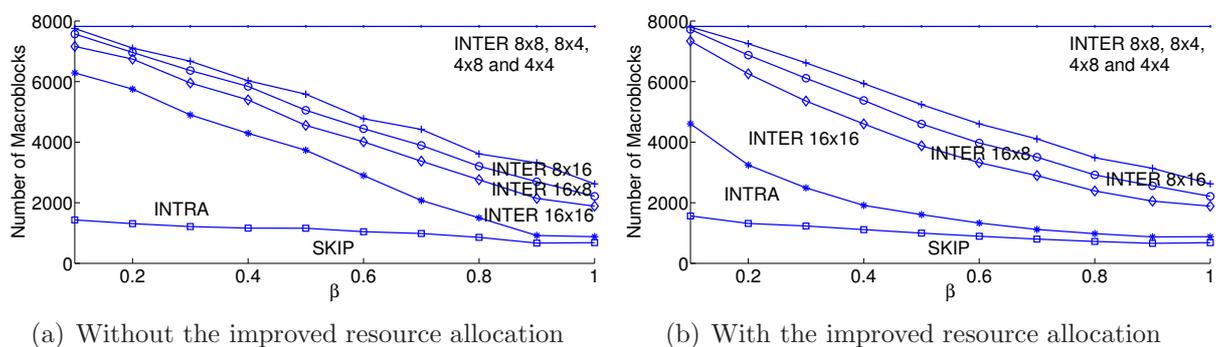
**Figure 5.14** PSNR and Bit-rate gains with improved resource allocation over different  $\beta$  values for Football sequence



**Figure 5.15** Total encoding time in seconds vs.  $\beta$  curves for the complexity scalable encoder with improved resource allocation.



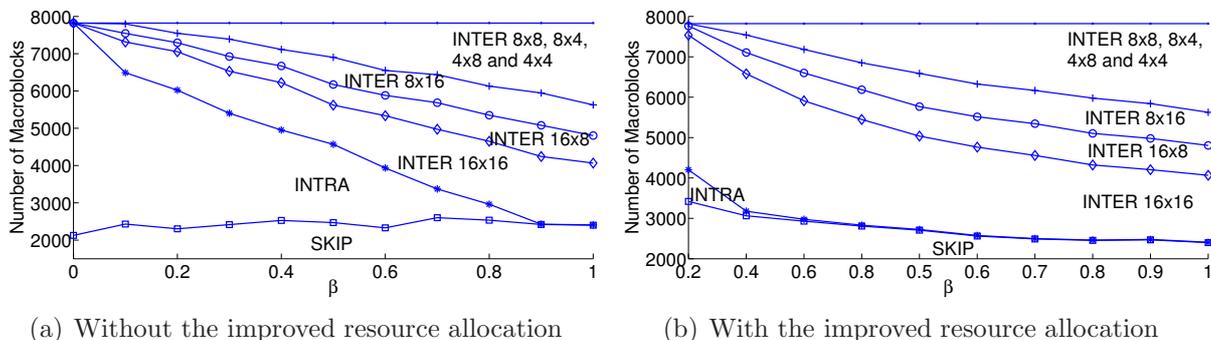
**Figure 5.16** RD curves with improved resource allocation for *Football*, each curve is comprised of 10  $\beta$  values ranging from 0.1 to 1.0 by increments of 0.1.



(a) Without the improved resource allocation

(b) With the improved resource allocation

**Figure 5.17** Mode decision comparison of with the improved resource allocation (b) and without the improved resource allocation (a) by distance-between-the-curves method between over different  $\beta$  values for *Football* sequence.



**Figure 5.18** Mode decision comparison of with the improved resource allocation (b) and without the improved resource allocation (a) by distance-between-the-curves method between over different  $\beta$  values for Foreman sequence.

## 5.4 Complexity Reduction Extension: Application of EST & EMT in Scalable Encoding

In the previous section, we employed the RD prediction algorithm in the original complexity scalable encoding framework and modified the resource allocation system. The original system was solely based on the present RD costs of macroblocks as it delegated the RD operations to the macroblocks with largest RD costs in the wavefronts. However, the new system recognizes that the RD costs of different macroblocks may vary largely and follows a more conscious and smarter logic. It does not simply try to reduce the RD costs of macroblocks but rather it tries to attain the predicted values. Correspondingly, simulation results showed significant improvements in terms of both PSNR and bit-rate. Hence, from this point on, the term “complexity scalable encoding framework” (CSEF) will include the improved resource allocation technique unless specified otherwise.

In this section, we will incorporate the two complexity reduction tools, Early SKIP Termination (EST) and Early Mode Termination (EMT), that were introduced in Chapter 4, in CSEF and will explain the different implications that they have on the operation of the complexity scalable encoder. As its name suggests, EST tries the SKIP mode before any other mode and terminates the mode decision operation for a macroblock if its RD cost for SKIP mode is less than or equal to its predicted RD cost. In the complexity scalability concept, in addition to further reducing an encoder complexity that has already been limited by the control parameter  $\beta$ , EST will also eliminate the possible SKIP macroblocks

from the mode decision process so that there will not be any RD operations spent on these macroblocks. Hence, the RD operation budget will be distributed among the remaining MBs in the wave-front. The steps to the complexity scalable encoding framework with EST (CSEF-EST) are listed below and its block diagram is illustrated in Fig. 5.19. For each wave-front in the current frame:

1. Reset the counter to zero.
2. For each MB in the wave-front:
  - (a) Compute the predicted RD cost  $\hat{J}_i$  by Equation 4.1.
  - (b) Set the MB ‘Done’ flag,  $Done_i$  to FALSE.
  - (c) Evaluate SKIP mode.
  - (d) If  $J_i(SKIP)$  is less than or equal to  $\hat{J}_i$ , set the MB mode to SKIP and  $Done_i$  to TRUE, and move to the next MB in the wave-front.
  - (e) If not, evaluate INTRA modes.
  - (f) Store the mode with the minimum RD cost and its RD cost as the current best mode  $MODE_i^{best}$  and as the current best RD cost  $J_i^{best}$ .
3. Pick the MB satisfying  $(\max(J_i^{best} - \hat{J}_i)(AND)(Done_i == FALSE))$  as the current MB.
4. Increase the number of operations counter by 1.
5. Evaluate the mode given by the output of the function *NextModeToTest* whose input is the current best mode of the MB in progress and if all the modes are tried set  $Done_i$  to TRUE.
6. Update the current best mode with the newly tried one if the newly tried mode has a smaller RD cost.
7. Repeat steps from 3 to 7 until the counter is equal to  $N_{op}(WF_i, \beta)$  or the  $Done_i$  flag is TRUE for all MBs.



Early Mode Termination not only applies the early termination condition on SKIP mode but it applies it on the other modes as well. In scalable encoding with EST, when a macroblock that has an RD cost less than its predicted RD cost for an INTRA or an INTER mode but not for SKIP, the encoder does not take any special action and keeps evaluating the remaining modes as long as the RD operations budget allows it. However with EMT, the encoder will take such situations into consideration and will terminate the mode decision for macroblocks at the first mode satisfying the early termination condition. In other words, the encoder will continue trying different modes for a macroblock as long as its present RD cost is greater than its predicted RD cost. Therefore, CSEF-EMT is highly dependent on the accuracy of the RD cost prediction method. For instance, if the prediction method overshoots and estimates a value much higher than the optimal, it is highly probable that the encoder will terminate the mode decision for the particular macroblock at a mode with sub-optimal RD cost. However, the accuracy of our RD cost prediction method was tested extensively in Chapter 4 and the simulation results verified that using the predicted RD costs for early termination has negligible, if not minimal, impact on the RD performance of the encoder. The block diagram of the complexity scalable encoding framework with EMT is illustrated in Fig. 5.20 and the steps of the algorithm are as follows: for each wave-front in the current frame:

1. Reset the counter to zero.
2. For each MB in the wave-front:
  - (a) Compute the predicted RD cost  $\hat{J}_i$  by Equation 4.1.
  - (b) Set the MB ‘Done’ flag  $Done_i$  to FALSE.
  - (c) Evaluate SKIP mode.
  - (d) If  $J_i(SKIP)$  is less than or equal to  $\hat{J}_i$ , set the MB mode to SKIP and  $Done_i$  to TRUE, and move to the next MB in the wave-front.
  - (e) If not, evaluate INTRA modes.
  - (f) If  $J_i(INTRA)$  is less than or equal to  $\hat{J}_i$ , set the MB mode to INTRA and  $Done_i$  to TRUE, and move to the next MB in the wave-front.
  - (g) If not, store the mode with the minimum RD cost and the corresponding RD cost as the current best mode  $MODE_i^{best}$  and as the current best RD cost  $J_i^{best}$ .

3. Pick the MB satisfying  $(\max(J_i^{\text{best}} - \hat{J}_i)(AND)(Done_i == FALSE))$  as the current MB.
4. Increase the number of operations counter by 1.
5. Evaluate the mode M given by the output of the function *NextModeToTest* whose input is the current best mode of the MB in progress and if all the modes are tried set  $Done_i$  to TRUE.
6. If  $J_i(M)$  is less than or equal to  $\hat{J}_i$ , set the MB mode to M and  $Done_i$  to TRUE, and go to step 8.
7. If not, update the current best mode with the newly tried one if the newly tried mode has a smaller RD cost.
8. Repeat steps from 3 to 8 until the counter is equal to  $N_{op}(WF_i, \beta)$  or the  $Done_i$  flag is TRUE for all MBs.

#### 5.4.1 Experimental Results

It was shown in Chapter 4 that for some macroblocks, the predicted RD costs are slightly higher than their optimal values. Therefore, while encoding such macroblocks, the SKIP mode RD costs tend to be smaller than the predicted RD costs causing the early termination to kick in. In scalable encoding, we will observe the same trend to follow and regardless of the value of the control parameter  $\beta$ , both the EST and the EMT will cause an increase in the number of skipped macroblocks. In fact, we will see the number saturates for increasing  $\beta$  values. In return, this effect causes a slight saturation in the RD performances and a significant saturation in the encoding time which jeopardizes the linear relation between encoding time and  $\beta$ . In order to recapture the linearity, we redefine the RD operation budget Equation 5.1 and change the coefficient of 7 to 4. The reasoning behind the choice of 4 is that 4 gives a good cut-off point for which the preceding points do not cause a significant saturation in either the RD performance or the encoding time. The corresponding equation is shown below. Please note that we will demonstrate simulation results for both equations and will denote them by the equations used for the calculation of

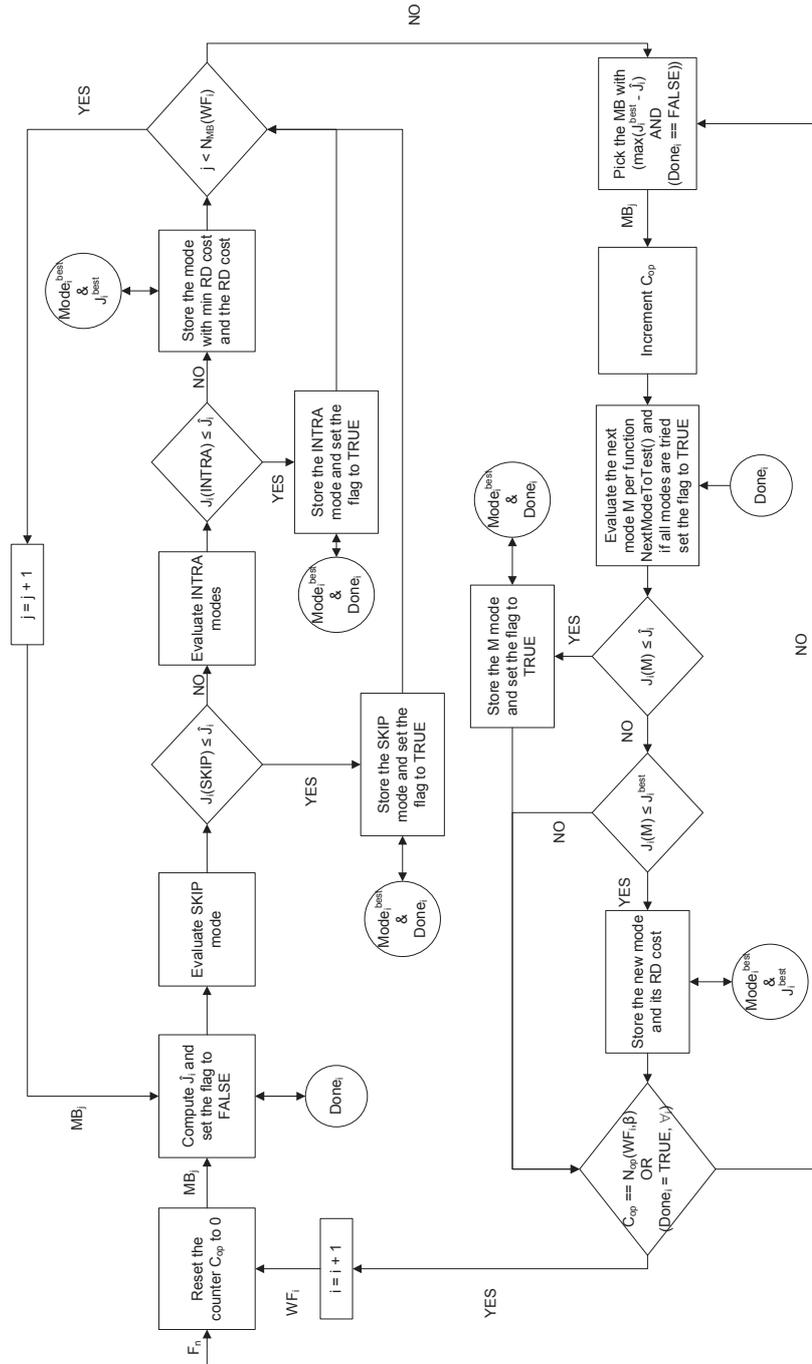


Figure 5.20 The Block diagram of The Complexity Scalable Encoding of Frame  $F_n$  with EMT.

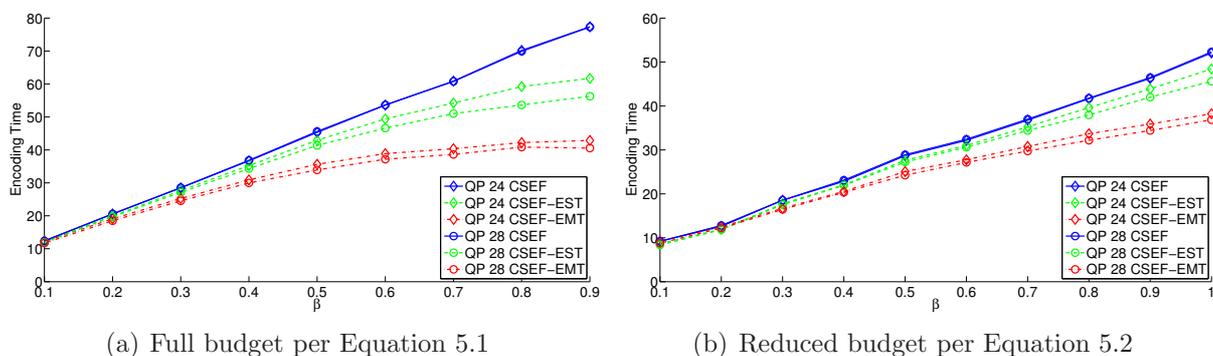
$N_{op}(WF_i, \beta)$ , i.e. Equation 5.1 for  $7 \times \beta$  (will also be referred as ‘full budget’) or Equation 5.2 for  $4 \times \beta$  (will also be referred as ‘reduced budget’).

$$N_{op}(WF_i, \beta) = 4 \times N_{MB}(WF_i) \times \beta, \quad (5.2)$$

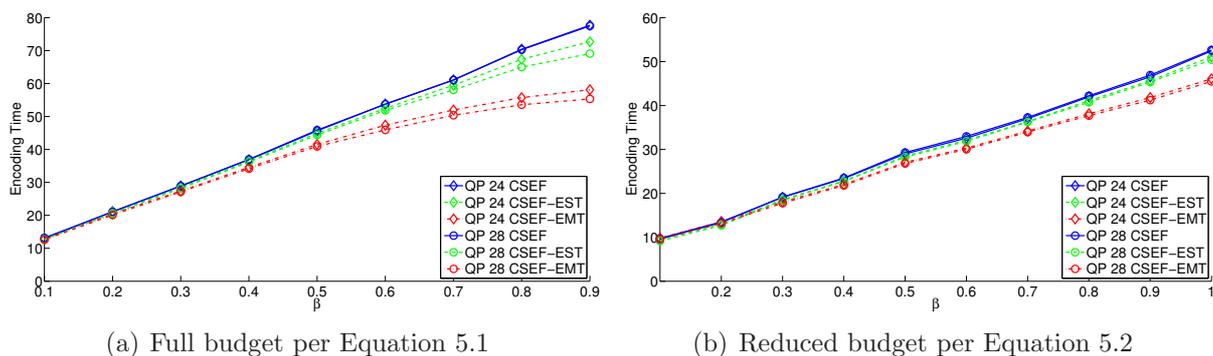
Figures 5.21 and 5.22 plot the encoding time against varying  $\beta$  values of Foreman and Football sequences for CSEF, CSEF-EST and CSEF-EMT. In the full budget scenarios, for both sequences, we observe the encoding time saturating as  $\beta$  increases in CSEF-EST and CSEF-EMT encoders. In the latter case, the saturation is quicker due to the stabilization of the number of macroblocks coded in INTER modes with large partition sizes, particularly INTER  $16 \times 16$ . In the CSEF and CSEF-EST encoders, all the INTER modes are on the decline except the INTER  $\leq 8 \times 8$  modes. This phenomenon is illustrated in Figures 5.27 and 5.29 for Foreman and Football sequences. In CSEF-EST encoding of the sequence, the stabilization in the number of SKIP coded macroblocks as opposed to the decaying in CSEF encoding is clearly seen. Additionally in CSEF-EMT encoding, we observe the stabilization of INTER  $16 \times 16$  as well. The same effect carries on to the RD curves. We can clearly see the RD points cluster together as  $\beta$  increases in Figures 5.23 and 5.25 for Foreman and Football sequences respectively.

When the RD operations budget is reduced per Equation 5.2, in Figures 5.21 and 5.22, we see that the linearity between the encoding time and  $\beta$  is preserved to some extent. Similarly, the RD points are more clearly separated from one another as demonstrated in Figures 5.24 and 5.26 for Foreman and Football sequences respectively. In accordance with these observations, in Figures 5.28 and 5.30, we see that the number of macroblocks coded in all modes except the INTER  $\leq 8 \times 8$  modes decline as  $\beta$  increases. If we compare the sequences and the QP values, we see that the clustering of RD points, the saturation of the encoding time and the mode decisions is more evident in Foreman, the slower sequence, and in higher QPs. As it was discussed in detail in Chapter 4, the reasoning of this is the fact that the motion in slow sequences and high QPs can be approximated with larger blocks; thus allowing encoder to rely heavily on SKIP and INTER  $16 \times 16$  blocks and less on INTER modes with small partition sizes.

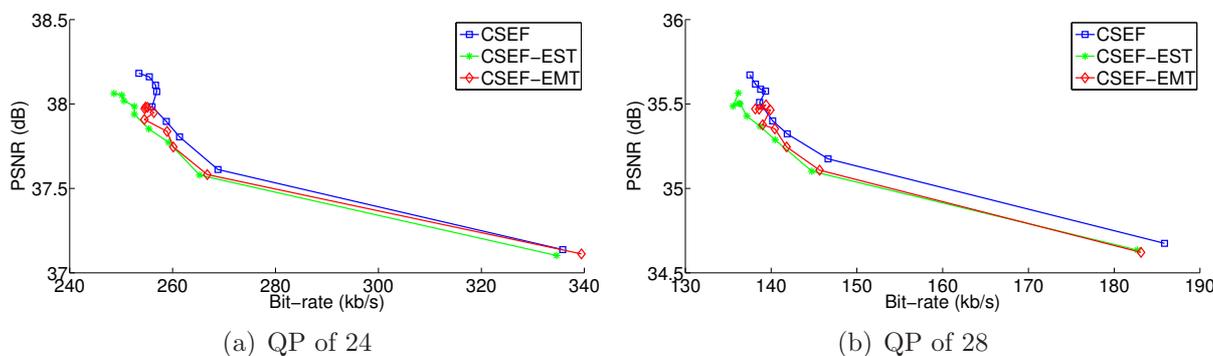
The analysis of the experimental findings bring us to several conclusions. The complexity scalable encoding framework (CSEF) has the ability to scale the encoder complexity up and down linearly through its control parameter  $\beta$ . As the encoder complexity varies, the



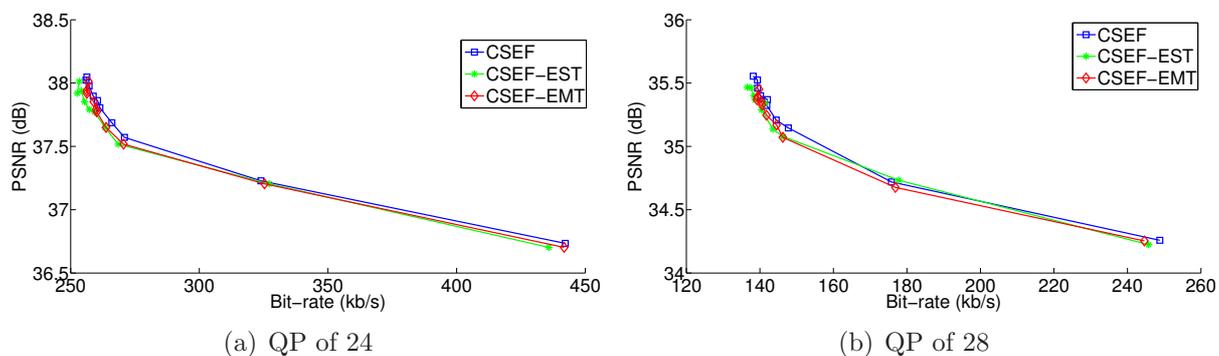
**Figure 5.21** Total encoding time in seconds vs.  $\beta$  curves for Foreman sequence with encoders: the complexity scalable encoding framework (CSEF), CSEF-EST and CSEF-EMT.



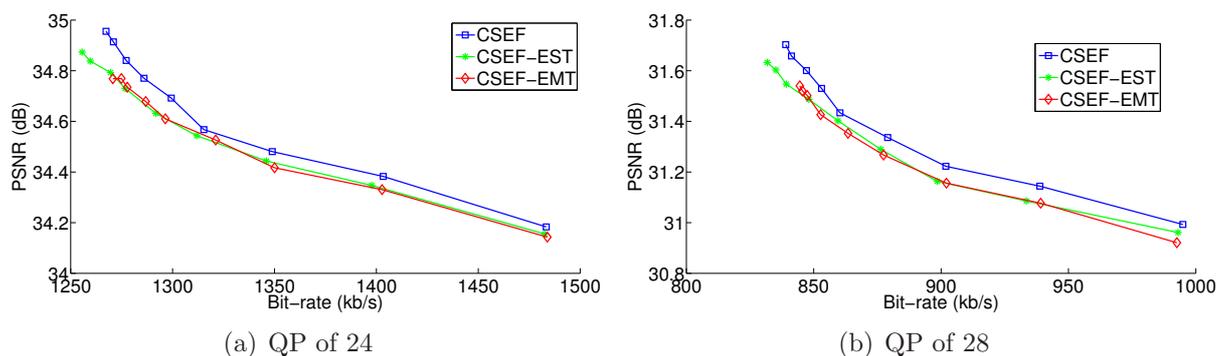
**Figure 5.22** Total encoding time in seconds vs.  $\beta$  curves for Football sequence with encoders: the complexity scalable encoding framework (CSEF), CSEF-EST and CSEF-EMT.



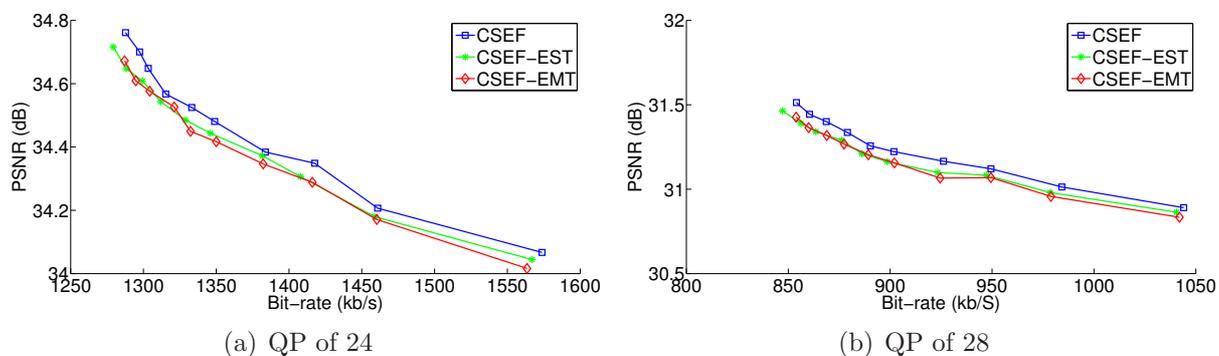
**Figure 5.23** RD curves of Foreman sequence with varying  $\beta$  points for full budget CSEF, CSEF-EST and CSEF-EMT encoders per Equation 5.1.



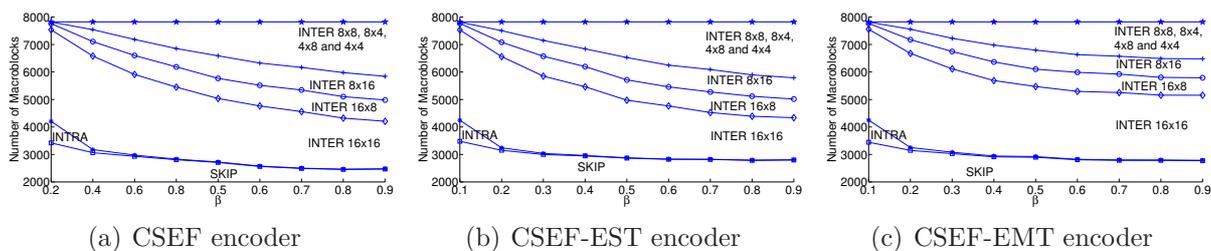
**Figure 5.24** RD curves of Foreman sequence with varying  $\beta$  points for reduced budget CSEF, CSEF-EST and CSEF-EMT encoders per Equation 5.2.



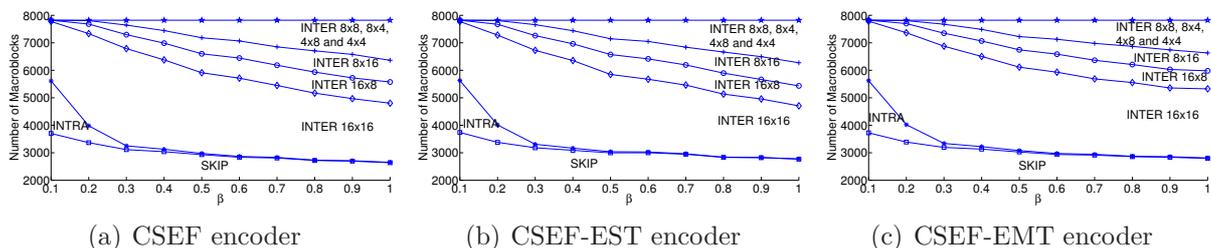
**Figure 5.25** RD curves of Football sequence with varying  $\beta$  points for full budget CSEF, CSEF-EST and CSEF-EMT encoders per Equation 5.1.



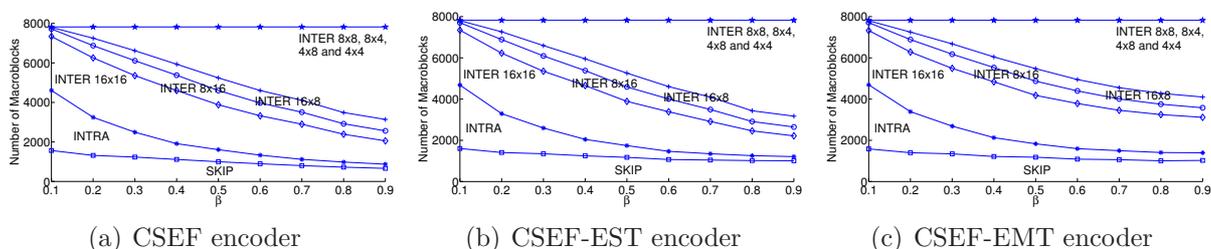
**Figure 5.26** RD curves of Foreman sequence with varying  $\beta$  points for reduced budget CSEF, CSEF-EST and CSEF-EMT encoders per Equation 5.2.



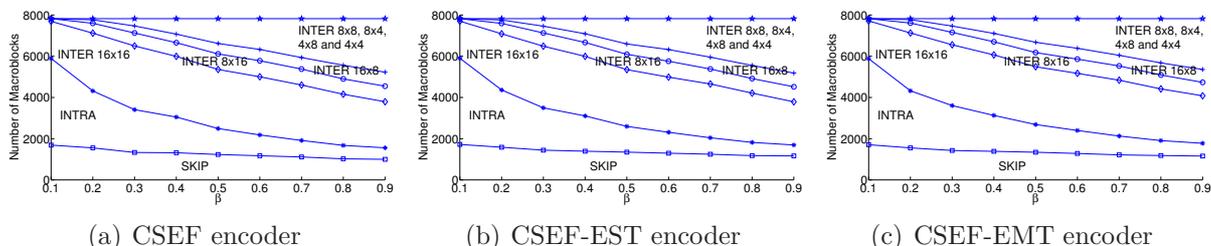
**Figure 5.27** Mode decision comparison by distance-between-the-curves method of full budget CSEF, CSEF-EST and CSEF-EMT encoders per Equation 5.1 against varying  $\beta$  points for Foreman sequence with a QP of 24.



**Figure 5.28** Mode decision comparison by distance-between-the-curves method of reduced budget CSEF, CSEF-EST and CSEF-EMT encoders per Equation 5.2 against varying  $\beta$  points for Foreman sequence with a QP of 24.



**Figure 5.29** Mode decision comparison by distance-between-the-curves method of full budget CSEF, CSEF-EST and CSEF-EMT encoders per Equation 5.1 against varying  $\beta$  points for Football sequence with a QP of 24.



**Figure 5.30** Mode decision comparison by distance-between-the-curves method of reduced budget CSEF, CSEF-EST and CSEF-EMT encoders per Equation 5.2 against varying  $\beta$  points for Football sequence with a QP of 24.

CSEF successfully adjusts the PSNR and bit-rate of the compressed bitstream. Including Early SKIP Termination (EST) and Early Mode Termination (EMT) in CSEF, boost the encoding speed, further reducing the encoder complexity, with a slight penalty in the RD performance. At the same time, since the complexity reduction rate is proportional to the  $\beta$  value (i.e. the greater the  $\beta$  value, the greater the complexity reduction rate), the linearity of the encoder complexity control is compromised. However, for scenarios where the RD operations budget is reduced or for the RD operations budget is full with a reduced range of, yet finer tuned  $\beta$  (e.g. 0.1 to 0.5 with steps of 0.05), the linearity is retained and the complexity is reduced. Therefore, in practice, CSEF can work with full budget (though high complexity) scenarios; but, in low complexity scenarios, in addition to its ability reduce the encoder complexity, it has the option to activate the EST or the EMT in order to reduce the complexity at greater extents.

## 5.5 Summary

This chapter presented a novel complexity scalable encoding framework. The notion of complexity scalability was defined and discussed in the context of video coding. The feasibility of such an algorithm in H.264/AVC encoding framework was examined and a macroblock level scalability was found to be conceivable with the so-called wave-front MB scheduling technique. Through this technique and our analysis of the encoder complexity, we designed the singly parameterized complexity scalable encoding framework (CSEF). It was shown that the complexity control of CSEF is superior to a similar work by Tan, Lee, Tham and Rahardja [23]. The RD prediction method from Chapter 4 was introduced in the CSEF as a means to improve the resource allocation of the encoder to the macroblocks. Simulation

results showed significant refinements in terms of both PSNR and bit-rate; exceeding 1dB and 50 percent respectively in many cases. Finally, the complexity reduction tools, Early SKIP Termination and Early Mode Termination, from Chapter 4 were also incorporated in the CSEF. The results showed additional reduction rates over different encoder complexities at the expense of slight loss in the RD performance. In the next chapter, we conclude this thesis and present some suggestions for future work.

# Chapter 6

## Conclusion

### 6.1 Summary

This thesis addresses the problem of computational complexity in software-only H.264/AVC video encoders. In the scenarios of real-time multimedia and mobile video, where the host devices are typically battery powered and limited in processing power, the encoder computational complexity becomes a key constraint. The achieved video quality and bit-rate of a coded sequence depend on the encoder complexity. The distribution of the complexity over different encoding tools is examined and the profiling test results showed that the mode decision process, which is carried out for each and every processing unit of a video, i.e. a macroblock, to obtain its coding mode, is the most computationally intensive operation in H.264/AVC encoding. The aim of this work is to provide novel tools and algorithms that limit and/or adaptively control the encoder complexity in order to attain the optimal rate-distortion (RD) performance.

A useful RD cost prediction method is described in Chapter 4. The method estimates the RD penalty of coding a macroblock with the output of the mode decision process accurately. The same chapter introduces two complexity reduction algorithms which employ the RD cost prediction method. The first of the two, is called the Early SKIP Termination (EST), uses the predicted RD costs to predetermine the SKIP macroblocks for which the algorithm omits the trials of INTRA and INTER modes. Simulations showed varying results depending on the input video characteristics. The encoding time savings proved to be significant for slower sequences which naturally contain a large number of SKIP mac-

roblocks. For faster sequence, the savings were found to be insufficient yet normal as such sequences rarely have any SKIP macroblocks. The second of the two complexity reduction algorithms, the Early Mode Termination (EMT), extends the EST algorithm and relies more heavily on the RD cost prediction method than EST. It processes the modes in an ascending complexity order starting with the SKIP mode and selects the first mode whose RD cost is less than or equal to the predicted RD cost. EMT achieves greater encoding time savings than EST at the cost of a slight decrease in the RD performance.

Chapter 5 described a novel complexity scalable encoding framework (CSEF) that can control the encoder complexity at a macroblock level through a single parameter. Results showed that CSEF has a superior complexity control in comparison to a similar work by Tan, Lee, Tham and Rahardja [23]. The RD cost prediction method was installed in the framework to improve the allocation of computational resources to macroblocks. Experimental results showed striking improvements in the visual quality and the bit-rate of the coded sequences regardless of the encoder complexity. As the last step, EST and EMT were incorporated and tested in the context of complexity scalability. Results showed that EST and EMT can provide the complexity scalable encoder with additional complexity reduction capabilities which are normally unattainable by the complexity control mechanism of CSEF.

The techniques described in this thesis are within the confines and the scope of this research work. They fulfill the primary objective of providing tools to manage the encoder complexity of the H.264/AVC video coding standard. The proposed complexity reduction and complexity scalability algorithms can alleviate the computational burden of H.264/AVC encoding in power and resource constrained systems, such as mobile phones and PDAs, with minimal loss in video quality.

## 6.2 Future Work

Low complexity and complexity scalable video encoding are notions with many possible solutions and there are several possible extensions to the techniques presented in this thesis. Both the proposed Early Mode Termination algorithm and the complexity scalable encoding framework (CSEF) try the coding modes in an ascending complexity order which is constant for all the macroblocks. While this order works well for slow moving sequences in which a large number of macroblocks are coded with low complexity modes, for faster sequences

in which macroblocks are usually coded with high complexity modes, it does not work as well. Therefore, a possible way around this could be to keep track of the coding modes of co-located macroblocks in the past frames. This information can be used to change the order of mode trials. For instance, if a macroblock has been coded in the past mostly with the SKIP mode or an INTER mode with large partitions (i.e.  $16 \times 16$  or  $16 \times 8$  or  $8 \times 16$ ), the original order can be kept. However, if a macroblock has been coded in the past as mostly with an INTER mode with small partitions (i.e.  $8 \times 8$  or  $8 \times 4$  or  $4 \times 8$  or  $4 \times 4$ ), the order may be changed to descending complexity where INTER  $8 \times 8$  and sub- $8 \times 8$  modes will be tried prior to INTER  $16 \times 16$ , INTER  $16 \times 8$  and INTER  $8 \times 16$ . In the context of low complexity encoding, this practice would allow greater encoding time savings and in the context of complexity scalable encoding, it would improve the obtained visual quality and bit-rate values for particularly faster sequences.

The simulations presented in this work have used the Full Search Algorithm for motion estimation. However, using different search algorithms would change the complexities of different mode trials. Therefore, another possible extension to this work could be to modify the mode decision order and to alter the RD operations assigned for each mode trial in accordance with the new complexity granularity. A third extension could be to model the PSNR and the bit-rate against the control parameter of CSEF. Bit-rate is another key constraint in data compression and in many video compression applications the visual quality is adjusted per the available bandwidth. While the complexity modeling of CSEF was established in the discussion of the framework, the PSNR or the bit-rate modeling was not covered. The analysis of the behaviors of PSNR and bit-rate in CSEF with respect to the control parameter could formulate a more accurate and practical optimization problem.

---

## Bibliography

- [1] Y. H. Tan, W. S. Lee, J. Y. Tham, and S. Rahardja, “Complexity rate distortion optimization for real-time H.264/AVC encoding,” in *Proc. IEEE ICCCN’09*, San Francisco, CA, USA, Aug. 2009.
- [2] I. E. G. Richardson, *H.264 and MPEG-4 Video Compression : Video Coding for Next-Generation Multimedia*, 1st ed. John Wiley & Sons Ltd, 2003.
- [3] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, “Overview of the H.264/AVC video coding standard,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–575, Jul 2003.
- [4] A. Ortega and K. Ramchandran, “Rate distortion methods for image and video compression,” *IEEE Signal Process. Mag.*, vol. 15, no. 6, pp. 23–50, Jan 1998.
- [5] G. J. Sullivan and T. Wiegand, “Rate distortion optimization for video compression,” *IEEE Signal Process. Mag.*, vol. 15, no. 6, pp. 74–90, Nov 1998.
- [6] C. S. Kannangara, I. E. Richardson, and A. J. Miller, “Computational complexity management of a real-time H.264/AVC encoder,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 9, pp. 1191–1200, Sep 2008.
- [7] K. R. Rao and P. Yip, “Discrete Cosine Transform: Algorithms, advantages, applications,” San Diego, uppercaseCA: Academic Press, 1990.
- [8] I. E. G. Richardson and Y. Zhao, “Adaptive management of video encoder complexity,” *Real-Time Imaging*, vol. 8, no. 4, pp. 291–301, Aug 2002.
- [9] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, “Motion compensated interframe coding for video conferencing,” in *Proc. National Telecommunication Conference’81*, New Orleans, Louisiana, 1981, pp. G5.3.1–G5.3.5.
- [10] J. R. Jain and A. K. Jain, “Displacement measurement and its application in inter-frame image coding,” *IEEE Trans. Commun.*, vol. 29, no. 12, pp. 799–1806, Dec 1981.

- 
- [11] M. Ghanbari, "The cross-search algorithm for motion estimation," *IEEE Trans. Commun.*, vol. 38, no. 1, pp. 950–953, Jul 1990.
- [12] J. Y. Tham, S. Ranganath, M. Ranganath, and A. A. Kassim, "A novel unrestricted center-biased diamond search algorithm for block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 4, pp. 369–377, Aug 1998.
- [13] Z. Chen, P. Zhou, and Y. He, "Fast integer and fractional pel motion estimation for JVT," JVT of ISO/IEC MPEG and ITU-T VCEG 6th Meeting, Awaji Island, Japan, Tech. Rep. JVT-F017, Dec. 2002.
- [14] K. P. Lim, G. Sullivan, and T. Wiegand, "Text description joint model reference encoding methods and decoding concealment methods," JVT of ISO/IEC MPEG and ITU-T VCEG, Munich, Germany, Tech. Rep. JVT-K049, Mar. 2004.
- [15] X. Yi, J. Zhang, N. Ling, and W. Shang, "Improved and simplified motion estimation for JM," JVT of ISO/IEC MPEG and ITU-T VCEG, 16th Meeting, Poznan, Poland, Tech. Rep. JVT-P021, Jul. 2005.
- [16] A. M. Tourapis, H. Y. Cheong, and P. Topiwala, "Fast ME in the JM reference software," JVT of ISO/IEC MPEG and ITU-T VCEG, 16th Meeting, Poznan, Poland, Tech. Rep. JVT-P026, Jul. 2005.
- [17] S. Wu, K. P. Lim, D. J. Wu, S. Rahardja, X. Limn, F. Pan, and Z. G. Li, "Fast mode decision for intra prediction," JVT of ISO/IEC MPEG and ITU-T VCEG 7th Meeting, Pattaya II, Thailand, Tech. Rep. JVT-G013, Mar. 2003.
- [18] K. P. Lim, S. Wu, D. J. Wu, S. Rahardja, X. Limn, F. Pan, and Z. G. Li, "Fast inter mode selection," JVT of ISO/IEC MPEG and ITU-T VCEG 9th Meeting, San Diego, USA, Tech. Rep. JVT-I020, Sep. 2003.
- [19] B. Jeon and J. Lee, "Fast mode decision for H.264," JVT of ISO/IEC MPEG and ITU-T VCEG 10th Meeting, Waikoloa, Hawaii, USA, Tech. Rep. JVT-J033, Dec. 2003.
- [20] C. S. Kannangara, I. E. G. Richardson, M. Bystrom, J. R. Solear, Y. Zhao, A. Maclennan, and R. Cooney, "Low complexity skip prediction for H.264 through Lagrangian cost estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 2, pp. 202–208, Feb 2006.
- [21] R. Braspenning, G. de Haan, and C. Hentschel, "Complexity scalable motion estimation," in *Proc. VCIP'02*, vol. 4671(1/2), San Jose, CA, USA, Jan 2002, pp. 442–453.

- 
- [22] M. Gallant, G. Cote, and F. Kossentini, “An efficient computation-constrained block-based motion estimation algorithm for low bit rate video coding,” *IEEE Trans. Image Process.*, vol. 8, no. 12, pp. 1816–1822, Dec 1999.
- [23] K. Lengwehasatit and A. Ortega, “A novel computationally scalable algorithm for motion estimation,” in *Proc. VCIP’98*, San Jose, CA, USA, Jan 1998, pp. 68–79.
- [24] [Online]. Available: [http://ip.hhi.de/imagecom\\\_G1/savce/downloads/SVC-Reference-Software.htm](http://ip.hhi.de/imagecom\_G1/savce/downloads/SVC-Reference-Software.htm)
- [25] [Online]. Available: <http://iphome.hhi.de/suehring/tml/>
- [26] S. B. Solak and F. Labeau, “Complexity scalable encoding for power-aware application,” The Work in Progress in Green Computing (WIPGC) Workshop of IGCC, Chicago, USA, Aug. 2010, accepted.