Department of Electrical and Electronic Engineering
Loughborough University

Mode Decision for the H.264/AVC Video Coding Standard

Mingyuan Yang

A thesis submitted in partial fulfillment for the award of
Doctor of Philosophy at Loughborough University

July 2006

Supervisor: Christos Grecos, PhD MIEEE
Research Director: Prof. David Parish
Applied Signal Processing Group

Acknowledgement

I am taking this opportunity to thank individually, all those who have assisted me in one way or the other with my Ph.D Project.

First of all, I would like to take this opportunity to extend my deepest gratitude to my supervisor, Dr. Christos Grecos, for his sharp eye on details and superb analytical skill, these have been instrumental in the success of my project. I have great respect for his kind attention and time.

I am also grateful to my colleague, Lihui Chen for his assistance and companionship throughout my stay at the University.

Secondly, I am greatly indebted to my family for their understanding, patience and support during the entire period of my study.

I also highly appreciate the opportunity extended to me to join Applied Signal Processing Group in Loughborough University and would like to extend my thanks to all staff in our group for their assistance in my research work.

Abstract

H.264/AVC video coding standard gives us a very promising future for the field of video broadcasting and communication because of its high coding efficiency compared with other older video coding standards. However, high coding efficiency also carries high computational complexity. Fast motion estimation and fast mode decision are two very useful techniques which can significantly reduce computational complexity.

This thesis focuses on the field of fast mode decision. The goal of this thesis is that for very similar RD performance compared with H.264/AVC video coding standard, we aim to find new fast mode decision techniques which can afford significant time savings. Firstly, fundamental concepts of video coding, background of H.264/AVC standard, mode decision schemes and fast mode decision techniques are introduced separately. Then, three of our main contributions on fast mode decision techniques are proposed explicitly. Inter mode decision scheme for P slices we propose firstly exploits neighbourhood information and a gentle set of smoothness constraints to predict inter modes. About 50% of time saving can be achieved compared with H.264/AVC standard. Then we propose a novel fast mode decision which can be considered as an extension of previous scheme. This scheme combines a set of conditions for Skip mode detection, two heuristics and monotonicity property. For very similar RD performance we achieve about 61% reduction in run times as compared to H.264/AVC standard. The third contribution for fast mode decision is improvement of fast inter mode decision technique which is adopted by H.264/AVC standard. It jointly exploits error and pixel information for fast mode prediction. In the end, limitations of the proposed fast mode decision schemes and directions for the future research are discussed.

Keywords: Video coding, H.264/AVC, Rate distortion optimization (RDO), Mode decision, Fast mode prediction, Complexity

Content

List of Figures

List of Tables

List of Graphs

Chapter 1

Introduction and Summary

1.1 Introduction to Video Coding

1.1.1 Overview of Video Coding

Digital video technology has been growing steadily in the last decade. New applications like video e-mail, third generation mobile phone with video communication capabilities, video conferencing, and video streaming on the web continuously push for further evolution of research in digital video coding. A problem, however, is that still image and digital video data rates are very large, for example for 10 seconds of uncompressed broadcast quality video (640x480 pixels, 24 frames/sec) they can even reach about 300Mbits. Data rates of this magnitude would consume a lot of the bandwidth, storage and computing resources in the typical personal computer. Hence, digital video compression techniques have played an important role in the world of telecommunication and multimedia systems such as MPEG-2 [29], MPEG-4 [30], H.263 [31] where the above factors are still valuable commodities.

Generally speaking, there is a large amount of spatial and temporal redundancy in digital video sequences. Video compression techniques exploit such redundancies for reducing the amount of information needed for representing picture sequences without losing much of quality, as judged by human viewers.

There are two categories of video coding techniques: lossless video coding and lossy video coding. Lossless video coding techniques compress video data without any information loss and during decompression an exact replica of the original data can be produced. Lossy video coding assumes some video information is not very important and can be discarded. Obviously compared with lossless video coding, lossy video coding can get a high compression ratio with some reduction of quality. Lossless and lossy video coding can be used in different applications based on their characteristics. For example, Medical imaging may require lossless video coding to make sure that doctors make the right diagnostic. Professional imaging needs to be stored in the original undistorted format for future processing. Also many high-end digital cameras enable the photographer to achieve the raw, uncompressed video sequences. However for video phones, video conferencing, web cameras or video broadcasting, some quality degradation is allowed and lossy video coding is a good option for these applications. In this thesis, we focus in the general cases which tolerate some quality loss. Therefore lossy video coding techniques will be described explicitly.

Block-Based Hybrid Video Coding is the most popular technique used in the lossy video coding systems. It is called block-based since each coded picture in video sequences is represented with block shaped units of associated luma and chroma samples, called macroblocks (16x16 pixel areas). It is called hybrid since it exploits both temporal correlation (inter coding) and spatial correlation (intra coding). Most of video coding standards which have been developed by the two organizations (International Organization of Standardization, International Electrotechnical Commission (ISO/IEC) and the International Telecommunications Union, Telecommunications Standardization Sector (ITU-T)) adopted a block-based hybrid video coding framework. (Note: There are some other new video coding standards like MPEG-4, MPEG-7 and MPEG-21 which adopt Object-Based hybrid Video Coding instead of Block-Based hybrid video coding for special purposes.) Due to the popularity of the block-based hybrid video coding approach, the fundamental concepts are introduced in section 1.1.2.

1.1.2 Fundamental Concepts of Block-based Hybrid Video Coding

A. General Video CODEC

A general block-based hybrid video codec (Figure 1-1) encodes a source video sequence into compressed video data and also decodes it to produce a copy (lossless compression) or approximation (lossy compression) of the original video sequence. On the encoder side, original video sequences are initially segmented into block or macroblock units and motion estimation/compensation is preformed for Inter frames in order to find the best motion vector(s) and the minimal residuals (the difference between the original video blocks and the best matches in the reference frame). The residual blocks are transformed, quantised, entropy coded and then sent to the network, while the motion vectors are entropy coded only and then transmitted. For intra frames, block or macroblock unites are transformed, quantised, entropy coded directly and sent to the network. The decoder part of the encoder and the decoder itself operate in a very similar manner. The only difference is that in the decoder the compressed residual errors or compressed video data are variable length decoded, but because the decoder part of the encoder is in front of the entropy coding obviously variable length decoding is not needed. Inverse quantised and inverse transformed to reproduce the reconstructed intra fames or the residuals which are added to the best matches from the reference frame(s) to produce the reconstruct inter frame. Such frames will be put into the delay buffer and then be used for motion estimation/compensation of future frames. As can be seen, on the decoder side, we have almost an exact inversion of each stage of the encoding process. The one stage that cannot be exactly inverted is the quantization stage due to information loss, so a reconstructed video data of original video sequence is obtained after the decoding process and played into the video player.

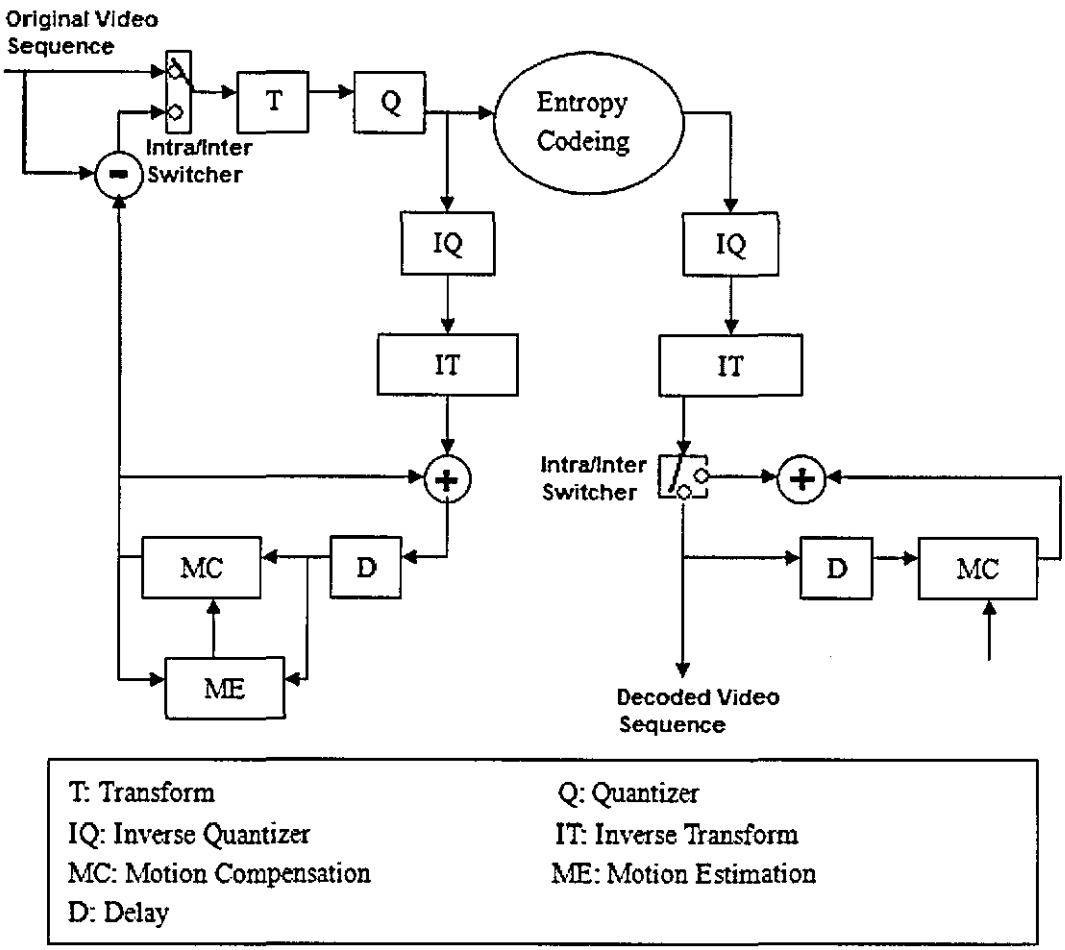


Figure 1-1 Block Diagram of a Video Codec

B. Motion Estimation and Motion Compensation

Successive video frames may contain the same objects (still or moving). Motion estimation examines the movement of objects in a video sequence and tries to obtain vectors representing the estimated motion. Motion compensation uses the information of the objects' difference between current frame and reference frame to achieve data compression. In inter frame coding, motion estimation and compensation have become powerful techniques to eliminate the temporal redundancy due to high correlation between consecutive frames. The simplest motion estimation and compensation method is the block-matching algorithm

(BMA), where the pixel values in blocks of each frame are estimated by a displaced block of similar shape and size in the past or future frame(s). The blocks are usually defined by dividing the video frame into block-wise units. Each block from current frame is matched into a block in the reference frame by shifting the current block over a predefined neighborhood of pixels in the reference frame. At each shift, the sum of the distances between the luma values of the two blocks in computed. The shift which gives the smallest total distance is considered the best match. Figure 1-2 illustrates a process of block-matching algorithm.

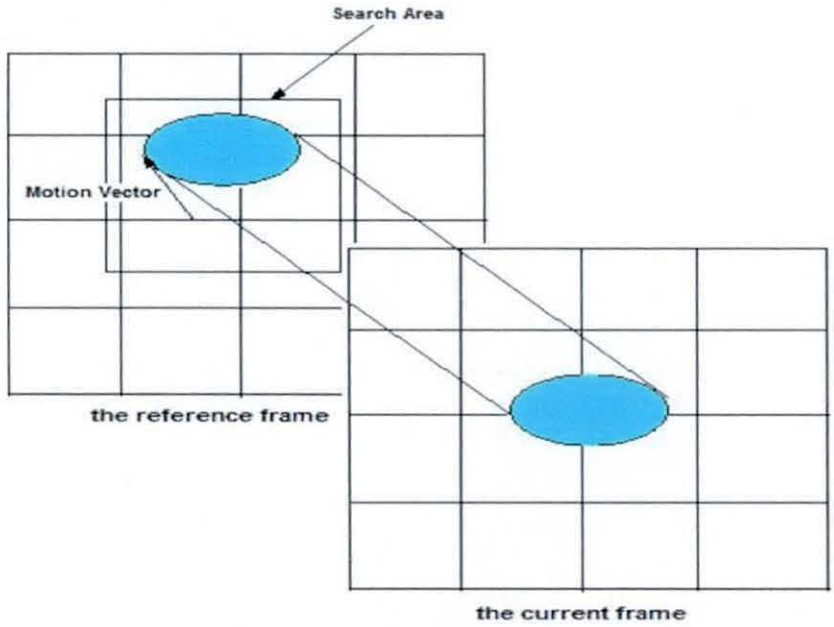


Figure 1-2 Process of Block-matching Algorithm

Generally there are three kinds of block-matching algorithms: Fixed Size Block-Matching (FSBM), Variable Size Block-Matching (VSBM) and Object based Block-Matching (OBM). Jain and Jain in [1] originally proposed the FSBM technique which is widely adopted because of its easy implementation. Each video frame is divided into a fixed number of fixed size block units. Each block searches in the reference frame to find the best match and produce the minimal residual. Block size is predefined in the video coding standards and the typical block sizes are of the 16x16 pixels. The choice of block-size to use for motion compensation is

always a compromise, since smaller and more in number blocks can better represent complex motion than fewer large ones. Ribas-Corbera and Neuhoff in [2] investigated this problem and found that the choice of the block-size can affect not only motion vector bits but also frame residual bits. Optimal choice of block size will achieve the best tradeoff between motion vector bits and frame residual bits. So researchers have proposed a number of Variable Size Block Matching algorithms to improve FSBM by varying the size of blocks which more accurately match moving areas. For example, Chan, Yu and Constantinides [3] have proposed a scheme that is called a top-down approach and starts with relatively large blocks, which are then repeatedly divided until a locally minimum error or the maximum number of blocks. For the same number of blocks per frame as FSBM, VSBM algorithm results in better prediction, thus achieving better tradeoff between the bits of motion vectors and bits of frame residuals. The performance of motion estimation and compensation can be further improved by using Object based Block-Matching (OBM). The idea behind OBM is to divide a frame in several objects according to their motion and edge characteristics rather than regular blocks, each object describing movement in a certain direction.

In order to reduce the computational complexity of full block-matching algorithms, many algorithms have been proposed to reduce the complexity of motion estimation which reduces the number of search positions in the reference frame(s). Such algorithms include Three Step Search (TSS) [4], Four Step Search (FSS) [5], the Cross Search Algorithm (CSA) [6], Spiral Search (SS) [7], the Hierarchical Search Block Matching Algorithm (HSBMA) [8], Hexagon-Based Search (HEXBS) [99], hybrid Unsymmetrical-cross Multi-Hexagon-grid Search (UMHexagonS) [100, 101] and so on. Fast motion estimation and compensation is not the focus of research of this thesis. Therefore the interested reader is referred to the above mentioned references.

The block-matching algorithms need matching criteria to find best matching blocks in the reference frame(s). Some of the most often used matching criteria such as Mean Absolute Distance (MAD) [9], Mean Squared Error (MSE), Sum of

Squared Error (SSE) [10] and Sum of Absolute Difference (SAD) [9] are mentioned here.

MAD is defined as: $$MAD(i,j) = (\frac{1}{MN})\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}|S(x,y) - C(x+i,y+j)| \quad (1\text{-}1)$$

MSE is defined as: $$MSE(i,j) = (\frac{1}{MN})\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}(S(x,y) - C(x+i,y+j))^2 \quad (1\text{-}2)$$

SSE is defined as: $$SSE(i,j) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1}(S(x,y) - C(x+i,y+j))^2 \quad (1\text{-}3)$$

SAD is defined as: $$SAD(i,j) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1}|S(x,y) - C(x+i,y+j)| \quad (1\text{-}4)$$

Where $S(x,y)$ and $C(x+i,y+j)$ represent the original and reconstructed pixel values in the relative position of (i, j) in the reference frame, M and N are length and width of the block unit.

There are also two other matching criteria: Sum of Absolute Transformed Difference (SATD) and Rate Distortion based measurements which will be described in chapter 3.

C. Transform Coding

After motion estimation and compensation, most of the temporal redundancy has been eliminated from the original video data. The residuals which are produced by subtracting the current original video blocks from the best matches in the reference frames (reconstructed previous/future frames) still contain spatial redundancy. Transform coding is used to convert the residual values to transform coefficient values. The desired effect is that most of the energy in the residual values will be transformed into a few large transform coefficients and most of spatial redundancy will be removed from the residuals in the transformed domain.

Many kinds of transforms have been tried for the video sequences, including for example Fourier Transform (FT) [11], Discrete Fourier Transform (DFT) [12], Karhunen–Loève Transform (KLT) [13], Discrete Wavelet Transform (DWT) [14], Discrete Cosine Transform (DCT) [15] and so on.

Discrete Cosine Transform (DCT) is conceptually similar to the DFT, except:

1) The DCT does a better job of concentrating energy into lower order coefficients than does the DFT for video frames.

2) The DCT is purely real, but the DFT is complex (magnitude and phase)

The above two advantages makes DCT to be the most popular transform coding technique which is widely adopted by most of video coding standards. Formulation of DCT and inverse DCT which is used to retrieve the spatial domain video values from their transformed domain is given below:

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

(1-5)

$$f(x,y) = \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u)C(v)F(u,v) \cos[\frac{(2x+1)u\pi}{2N}] \cos[\frac{(2y+1)v\pi}{2N}] \qquad (1\text{-}6)$$

In the above equation, u, v, x, y equals to 0, 1, 2, ...N-1; x, y are spatial coordinates in the sample domain; u, v are frequency coordinates in the transform domain; C(u), C(v) equals to $1/\sqrt{2}$ when u, v equals to 0; C(u), C(v) equals to1 otherwise; $f(x,y)$ and $F(u,v)$ are original signal values and transformed signal values.

After DCT transform coding, high values (lower frequency values) cluster around the top-left positions of the block matrix, while most of the remaining positions contain small or even zero values (high frequency values). Because our eyes are not very sensitive to the high frequency components, quantization can be used here to remove most of these small values and compression ratio will be even bigger.

D. Quantization

Quantization is a process of approximating the continuous set of values in the video data with a finite set of values. The input is original data, and the output is always one among a finite number of levels. Quantization makes it possible to represent the quantised signal with fewer bits than the original signal since the finite set of values is smaller. A good quantizer is one which represents the original signal with minimum loss or distortion.

There are two kinds of quantization, Scalar Quantization (SQ) [15] and Vector Quantization (VQ) [17]. Scalar Quantization is a mapping of an input value into a finite number of output values. If the input range is divided by an equal step size, the quantizer is a uniform quantizer. Bigger step size makes the set of output values smaller thus implying bigger distortion, on the contrary, smaller step size makes the set of output values bigger thus implying smaller distortion. An adaptive step size is a good choice for Scalar Quantization, this kind of Scalar Quantization is called Non-Uniform Quantization. Vector Quantization maps input data into a finite set of vectors. Each vector is called a code vector or a codeword and the set of all the code words is called a codebook. The advantage of Vector Quantization compared with the Scalar Quantization is bigger compression ratio, but both the encoder and decoder side need the codebook which make Vector Quantization more complex.

Because of the complexity problem, Scalar Quantization is often used in the current video coding standards. After quantization, continuous DCT coefficients become a set of finite values and a lot of zero coefficients are produced. After quantisation, Entropy Coding is performed on the coefficients.

E. Entropy Coding

After quantization and Zig-Zag scan [18], DCT coefficients change from two-dimensional space to one-dimensional space and most of zero DCT coefficients which are generated by quantization become neighbours each other. Entropy encoding is used here to reduce the statistical redundancy. Entropy

encoding is a coding method that assigns codes to symbols in order to match code lengths with the probabilities of the symbols. Typically, entropy encoders are used to compress data by replacing frequently-occurring values with shorter-length codes and replacing infrequently-occurring values with longer-length codes. The result is a sequence of variable-length codes. Because most common symbols use short codes, compression can be achieved.

Four of the most common entropy encoding techniques are Predictive Coding, Huffman Coding [19], Arithmetic Coding [20, 21] and Run Length Coding [22].

Predictive Coding is the simplest entropy encoding method which uses the previous symbols to predict current symbols. In video frames, symbols are highly correlated in local regions. For example, the quantization parameters and motion vectors of the macroblock to be encoded in relation to the previously encoded macroblock would not change too much. Coding efficiency can be improved by predicting such current parameters from previously-encoded neighbours and encoding the difference between the predicted and current values.

Huffman Coding uses a variable-length coding table for encoding a source symbol. Probability statistics of symbols need to be analysed before the coding process or during the coding process which is called dynamic huffman coding. Then as expected, short codes are assigned to the most frequently occurring symbols and long codes to less frequently occurring symbols.

Arithmetic Coding is a special kind of entropy coding. The main idea behind the arithmetic coding is the assignment to each symbol of an interval. Starting with the interval [0..1), each interval is divided in several subintervals, which sizes are proportional to the current probability of the corresponding symbols of the alphabet. The subinterval from the coded symbol is then taken as the interval for the next symbol. The output is a real number which is the interval of the last symbol. Arithmetic Coding produces near-optimal output for a given set of symbols and probabilities. Arithmetic coding typically has a better compression ratio than Huffman coding, as it produces a single symbol rather than several separate

codewords. But there are also some disadvantages, such as error robustness and high computation since several multiplications and divisions for each symbol are needed.

Run Length Coding is one of the simplest data compression algorithms but is effective for input symbols which consist of long sequences of repeated symbols. This repeated symbols is called a RUN, typically RLC encodes a run of symbols into two codes, a count (the number of repeating symbols) and a symbol.

There are some other entropy coding algorithms such as *Context Adaptive Variable Length Coding* (CAVLC) and *Context Adaptive Binary Arithmetic Coding* (CABAC) which are improvement of *Run Length Coding* and *Arithmetic Coding*. We will mention these algorithms in chapter 2.

1.1.3 Measurement of Video Quality

With the advent of various video compression standards, it has become increasingly important to devise a video quality measurement which can access the quality of different video coding products. There are two kinds of video quality measurements, objective and subjective measurements.

Many subjective video quality measurements are described in ITU-T recommendation BT.500 [23]. The main idea is that: video sequences are shown to some viewers and then their opinion is averaged to evaluate the quality of each video sequences. Subjective video quality measurements are more accurate measurements than objective ones because people are the final judges for video sequences. But subjective video quality measurements have two obvious disadvantages. Firstly, they are very tedious, thus cannot be performed in real time. Secondly, they are very difficult to be embedded into practical video coding systems because their automatic implementation is impossible.

Objective video quality measurements use some statistical features of the numerical errors between the distorted video frame and the reference frame.

Compared with subjective video quality measurements, objective measurements are simpler and easier to implement in the real time video coding systems. The two most widely adopted statistical measures are the Mean Squared Error (MSE) which is mentioned in section 1.1.2-B and the Peak Signal to Noise Ratio (PSNR). In practical video coding systems, PSNR is more popular than MSE in measuring the video quality, and its formula is illustrated below.

$$PSNR = 20 * \log_{10}((2^N - 1)/\sqrt{MSE}) \tag{1-7}$$

The disadvantage of both MSE and PSNR is that MSE and PSNR do not correlate well with subjective quality measures because the human perception of video distortions and artefacts is very complex.

The mainline stream of work to improve objective video quality measurements tries to put Human Visual System (HVS) [24] features into the objective measurements. Although HVS is too complex to fully understand with present psychophysical means, the incorporation of even a simplified model into objective measurements leads to a better correlation with the response of the human observers. Many algorithms such as Structural SIMilarity (SSIM) index have successfully employed HVS models [25, 26].

1.2 Summary of this thesis

In this thesis I restrict myself to the fast mode decision problem. Our aims and objectives are to find some new fast mode decision techniques which can achieve significant time savings without sacrificing too much RD performance compared with H.264/AVC video coding standard. Combining with other computational complexity reduction technique like fast motion estimation, we hope H.264/AVC standard can be used in real-time video coding systems like video conference and mobile video communication.

1.2.1 Chapter 2

Chapter 2 is a background introduction of H.264/AVC video coding standard. Compared to those older video coding standards H.264/AVC standard achieves high coding efficiency, but also carries high computational complexity because of its new features. In this chapter we will highlight these features which increase the complexity of H.264/AVC. An experiment of complexity analysis is performed in the end which can make us understand the complexity problem more clearly and easily.

1.2.2 Chapter 3

Chapter 3 is a literature review of existing fast mode decision approaches. In the chapter, we will discuss and compare various approaches to the fast mode decision problem in categories, including skip mode detection, fast inter mode prediction, fast intra mode prediction and fast intra/inter mode selection. The comparison will enable us to better understand the assumptions, advantages and limitations of these approaches.

1.2.3 Chapter 4

In this chapter, we present a novel layered inter mode prediction scheme for P slices in the H.264/AVC video coding standard. Our scheme exploits neighborhood information jointly with a set of skip mode conditions for enhanced skip mode decision, and also provides a set of smoothness constraints for fast inter mode decision. For similar RD performance as the standard, gains of in the range of 35-58% for run times and 33-55% for CPU cycles are observed as compared to the standard.

1.2.4 Chapter 5

We propose a novel framework for fast mode decision in the simple and main profiles of the H264 video coding standard. Our framework consists of a specific

combination of algorithms, each achieving computational savings while retaining Rate Distortion (RD) performance very similar to the standard. In particular, we utilise a set of skip mode conditions for P and B slices, two heuristics that reduce the cardinality of the inter mode set to be examined, inter/intra mode prediction and the monotonicity property of the Rate Distortion cost functions. We achieve content dependent savings in run times between 33 and 90.1% as compared to H264/AVC standard.

1.2.5 Chapter 6

A new algorithm for fast mode decision in the H.264/AVC video coding standard is presented in this letter. The algorithm exploits temporal and spatial information with adaptive thresholds and can provide significant computational savings with similar Rate Distortion performance as compared to accepted standard contributions [67, 70].

1.2.6 Chapter 7

This brings us to the end of this thesis. The conclusion chapter would summarize the contribution of fast mode prediction techniques we propose in this thesis. Limitations of our proposed schemes and possibilities for extension and improvements of the work are also discussed.

Chapter 2

Background of H.264/AVC Video Coding Standard

2.1 Introduction of H.264/AVC

The H.264/AVC video coding standard [27] is the newest video coding standard which is proposed by JVT (Joint Video Team) a union of ITU-T Video Coding Experts Group and ISO/IEC Moving Pictures Experts Group. H.264/AVC standard was finalized in May 2003 and the JVT also completed two generations of "corrigendum" errata corrections to the text of the standard. The new features it offers as compared to older video coding standard (for example H.263, MPEG2 and so on) enables H.264/AVC to achieve high coding efficiency but also carries high computational complexity. In this chapter, the new features and the complexity implications of H.264/AVC will be discussed.

The main purpose of H.264/AVC standard is to enhance compression performance and provide a "network-friendly" video representation addressing "conversational" (i.e., video telephony) and "non-conversational" (i.e., storage, broadcast, or streaming) applications. The structure of H.264/AVC standard is illustrated in Figure 2-1, which shows the typical video coding/decoding chain (excluding the transport or storage of video signal).

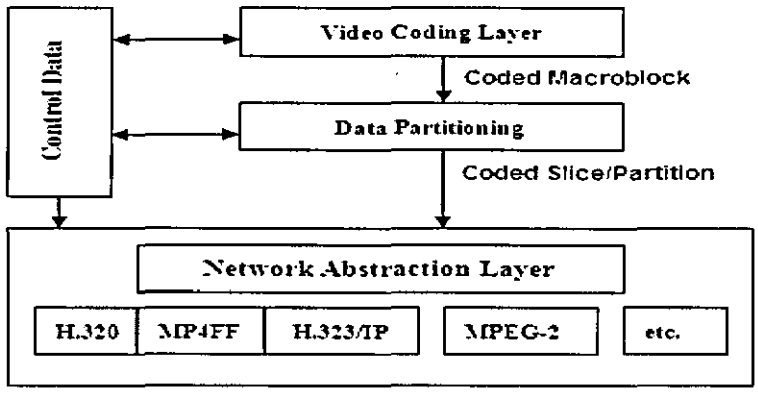


Figure 2-1 Structure of H.264/AVC Video Encoder

The Video Coding Layer (VCL) is used to compress the original video sequences into compressed video data by removing spatial and temporal redundancy. After VCL and Data Partitioning, compressed video data and some control data are sent to Network Abstraction Layer (NAL). NAL maps compressed video data to a generic format suitable for transport layer. In this thesis, we will not talk about the network related features of H.264/AVC, but we can refer the interested reader to [28].

H.264/AVC defines a set of three Profiles (Baseline, Main, Extended profile), each supporting a particular set of coding functions and each specifying what is required of an encoder or decoder that complies with the Profile. Different Profiles are used for different applications. Potential applications of the Baseline Profile include video-telephony, videoconferencing and wireless communications; potential applications of the Main Profile include television broadcasting and video storage; and the Extended Profile may be particularly useful for streaming media applications. Figure 2-2 shows the relationship between the three Profiles and the coding tools supported by the standard.

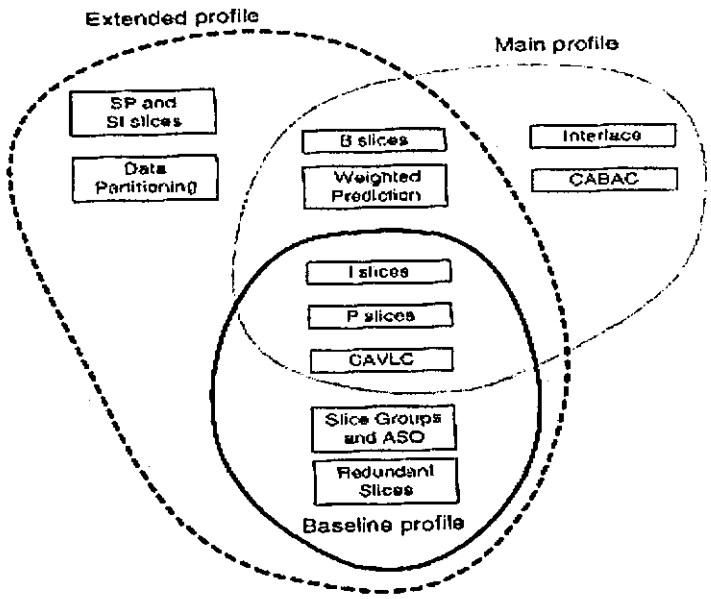


Figure 2-2 H.264 Baseline, Main and Extended Profiles

At a basic overview level, the coding structure of this standard is similar to that of all prior major digital video standards (H.261, MPEG-1, MPEG-2 / H.262, H.263 or MPEG-4 part 2). The architecture and the core building blocks of H.264/AVC encoder are similar to Figure 1.1 indicating that it is also based on motion-compensated DCT-like transform coding. However its main new features that make it stand out from previous standards (MPEG-2, MPEG-4, H263 [29-31] and derivatives etc.) are the great variety of applications in which it can be used and its versatile design. The main characteristics of H.264/AVC are summarised below:

1) High coding efficiency: Compared with MPEG-4, H.263+ and MPEG-2, H.264/AVC can achieve about 40%-60% bitrate saving at the same video quality.

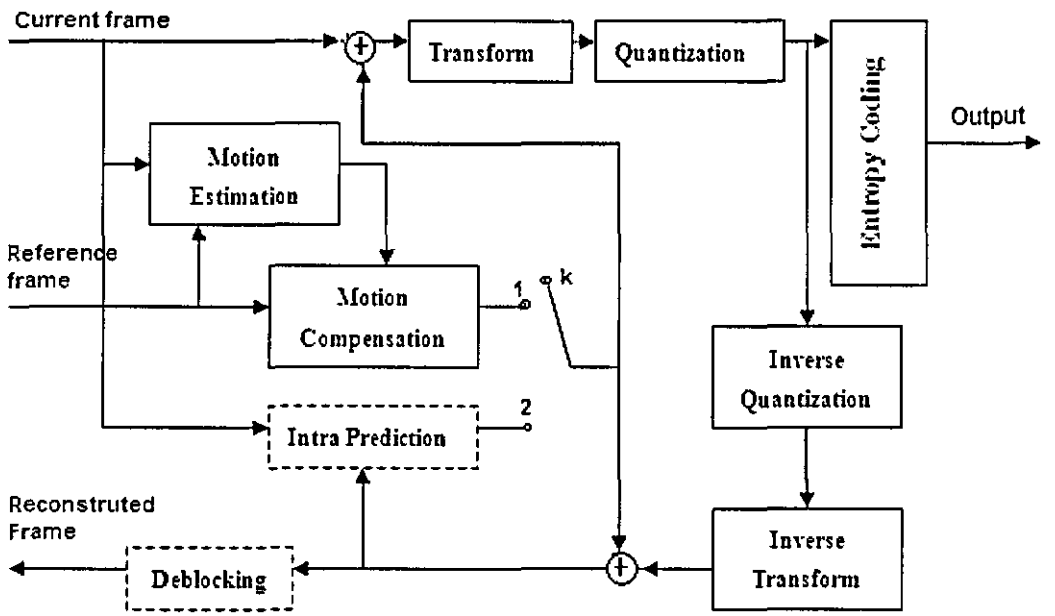
2) High video quality: H.264/AVC can provide high video quality in a wide range of bitrates (60Kbits/sec to 960Mbits/sec).

3) Adaptive delay: H.264/AVC not only can be used in low delay applications,

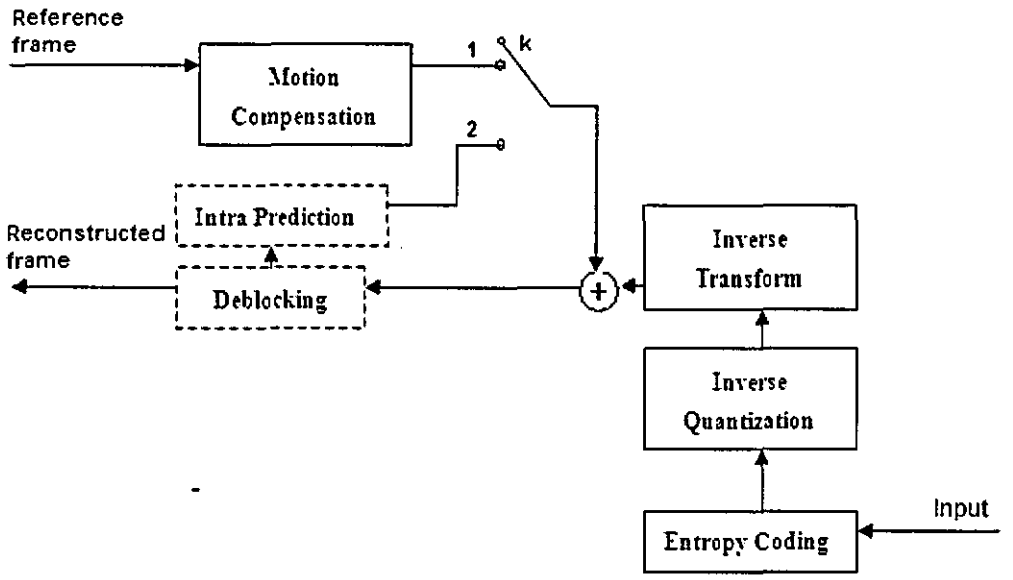
such as video conference, but also in applications without delay limitation, such as video storage and video streaming.

4) New error robustness tools: H.264/AVC provides new error robustness tools such as Parameter set structure, Flexible macroblock ordering (FMO), Arbitrary slice ordering (ASO), Redundant pictures, Data Partitioning and so on [27], which make it work well in high error-prone environments.

Figure 2-3 are block diagrams of H.264/AVC encoder and decoder used in the video coding layer. It has some similarities to figure 1-1, but the dotted rectangles are new modules of H.264/AVC encoder which are different from previous encoders such as that of the H.263 standard.



a)



b)

Figure 2-3 Block Diagrams of H.264/AVC CODECS

a) Encoder b) Decoder (k is a switcher of inter/intra prediction)

2.2 New Features of H.264/AVC

H.264/AVC standard has shown significant rate distortion improvements as compared to other standards for video compression. Figure 2-4 in [102] shows the luminance PSNR versus average bit rate for different coding standards, measured for the test sequence Paris CIF (100 frames). For video streaming applications, H.264/AVC BP (Baseline Profile), MPEG-4 SP (Simple Profile), H.263 CHC (Conversational High Compression), and H.263 Baseline Profile are considered. From the figure, we can see that based on these new techniques, H.264/AVC can achieve very good coding efficiency and quality performance compared with other previous standards.

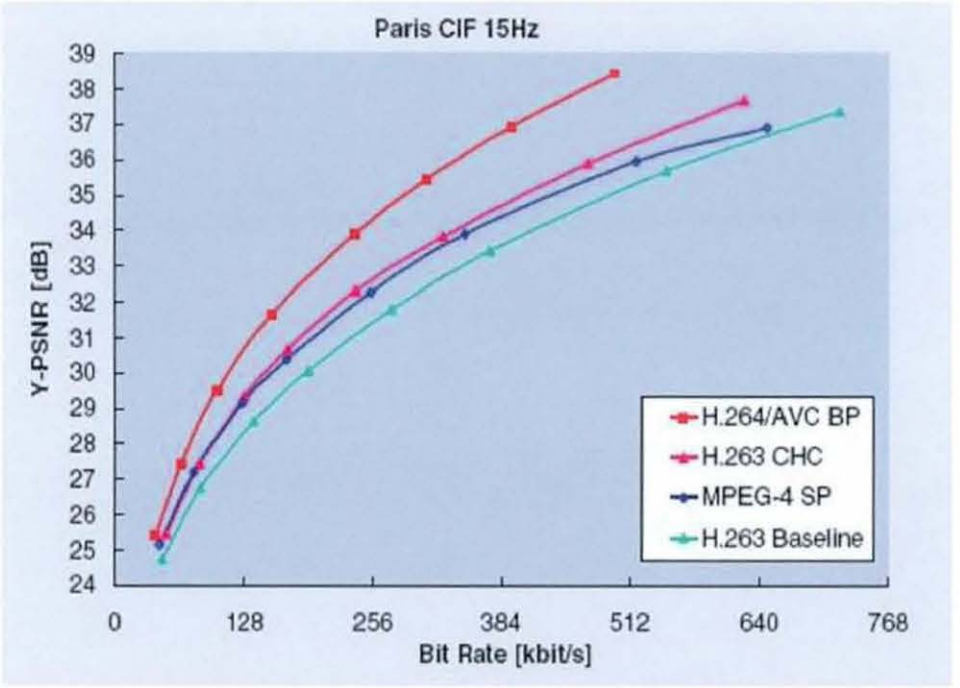


Figure 2-4 Luminance PSNR versus average bit rate for different coding standards, measured for the test sequence Paris CIF for video streaming applications

In this section we highlight some of its design features that enable such performance, as well as other features that enable the coding process in a more general sense.

A. Variable block size motion compensation

In the H.263 standard, a single video frame is divided into number of 16x16 macroblocks. Each macroblock includes four 8×8 luma blocks and two corresponding 8×8 chroma blocks, as shown in Figure 2-5. Motion estimation is performed on 8x8 block size.

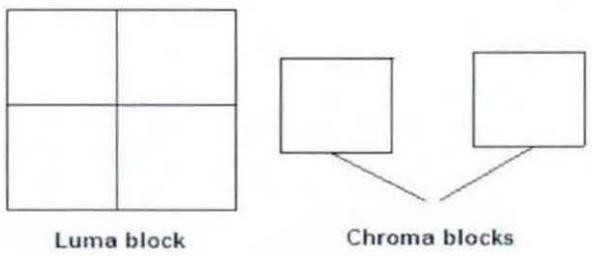


Figure 2-5 Macroblocks of H.263

H.264/AVC allows more flexibility in the size and shape of the motion compensation blocks than any other standard. Motion compensation on each 16x16 macroblock can be performed with various block sizes and shapes illustrated in Fig 2-6. The partitioning choice of a MB into 16x16, 8x16, 16x8 or 8x8 blocks is determined by macroblock types. In 8x8 mode each of sub blocks can be further divided independently into 8x8, 8x4, 4x8 or 4x4 sub-partitions determined by sub macroblock types. Its minimal luma motion compensation block size is 4x4.

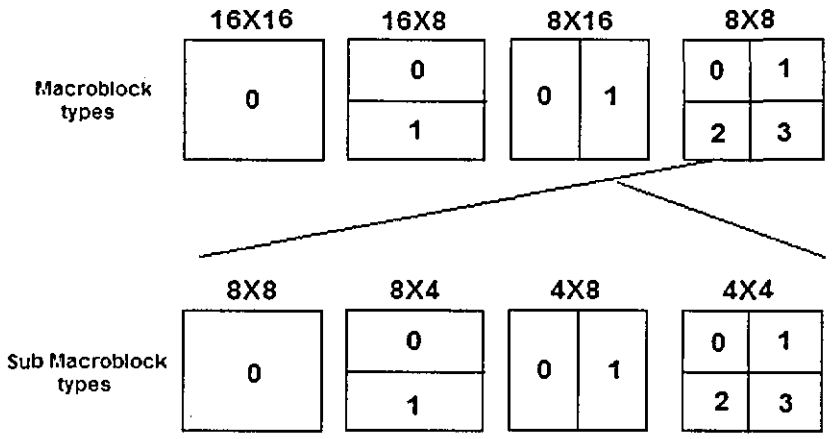


Figure 2-6 Different Macroblock Types in H.264/AVC

B. Quarter Sample Accurate Motion Compensation

MPEG-2 and H.263 permit integer-pixel and half-pixel accuracy motion estimation. Half sample pixels are calculated by 2-point linear interpolation from neighbour integer pixels. The accuracy of motion estimation is increased from the half sample motion vector accuracy to quarter sample motion vector accuracy in H.264/AVC. Each half sample pixels are calculated by a 6-tap Finite Impulse Response (FIR) filter [32] with weights (1/32, -5/32, 5/8, 5/8, -5/32, 1/32) from neighbour integer pixels. The 6-tap interpolation filter is relatively complex but produces an accurate fit to the integer-sample data and hence good motion compensation performance. Once all half sample pixels are available, quarter sample pixels can be calculated by 2-point linear interpolation from neighbouring integer and half sample pixels. Fractional sample positions are illustrated in figure 2-7, where capital letters represent integer positions, numbers represent half

sample positions and lower case letters represent quarter sample positions.

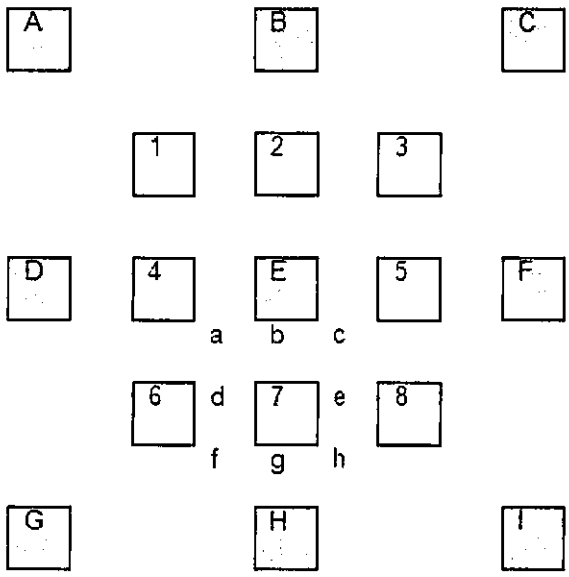


Figure 2-7 Fractional Sample Search Positions in H.264/AVC

After integer motion estimation, the best integer position is found. Assume that this integer position is E. Subsequently, the half sample positions 1, 2, 3, 4, 5, 6, 7, 8 are searched. Assume that position 7 is the best half sample position. Finally, the quarter sample positions a, b, c, d, e, f, g, h are searched.

C. Motion Vectors outside Picture Boundaries

While in previous standards, motion vectors had to be clipped inside reference picture boundaries, in H.264/AVC reference pictures are extrapolated instead (as first used as an optional feature of H.263), thus resulting in better quality block matches. In this feature, edge pixels are repeated as prediction of the points outside the boundaries of video frame. Significant gain can be achieved for movements along the edge of video frames, especially for small video formats such as 16 Common Intermediate Format (CIF), 4CIF or even smaller.

D. Multiple Reference Picture Motion Compensation

Instead of selecting only one or two reference pictures for motion compensation in previous standards, H.264/AVC standard allows multiple reference picture motion compensation from a number of frames in the buffer store, thus improving the coding efficiency but increasing the memory and computational complexity.

E. Decoupling of Referencing Order from Display Order

H.264/AVC allows a great degree of flexibility for choosing reference and display orders (bound only by the memory capacity of the system), which *was not the case* in older standards. A positive side effect of this flexibility is the removal of extra delay associated with bi-predictive coding which makes H.264/AVC not only suitable for high-delay applications but also for low-delay applications.

F. Bi-directionally Predicted Pictures can be used as References

In prior standards, bi-directionally-predicted pictures can not be used as references to predict other pictures in video sequences. The absence of this feature in previous standards, could lead to the inability of utilising a reference picture that may have been a closer approximation to the one to be encoded, thus leading to worse motion compensation.

G. Weighted Offsetting of Prediction Signal

This feature did not exist in previous standards. The best match signals found by motion estimation can be weighted and offset by weighted parameters which are predefined in the encoder and decoder. Coding efficiency can be improved especially for video sequences including fades. This feature can also be used for other purposes such as video effects.

H. Improved "Skipped" and "Direct" Mode Inference

H.1 Skip Mode

Before we describe the skip mode, the concept of "predicted motion vector" needs to be explained firstly. Due to high correlation among the motion vectors of adjacent blocks, we can predict the motion vector(s) of the current block from vectors of nearby, previously coded blocks and only send the different between the current motion vector and the predicted motion vector (MVD: Motion Vector Difference). In this manner we not only can speed up motion estimation but also can save a significant number of bits especially at low bit rates. Figure 2-8 and equation 2-1 illustrate the calculation process of prediction motion vector.

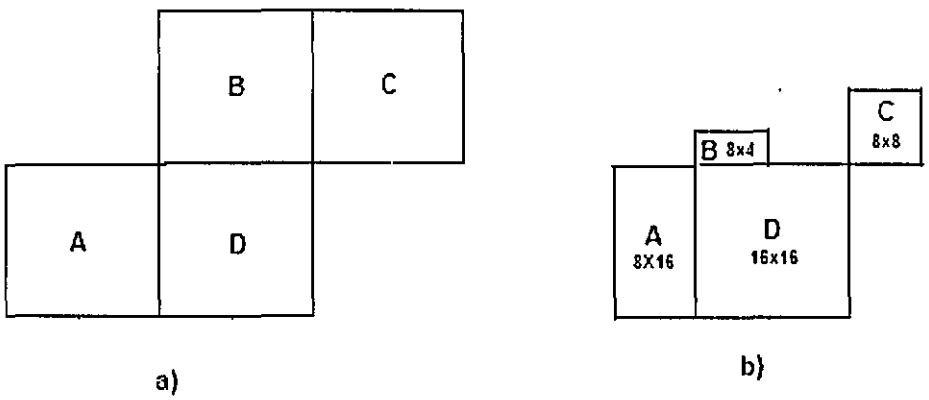


Figure 2-8 The Calculation Process of Prediction Motion Vector
a) Current block and neighbour blocks has the same partition
b) Current block and neighbour blocks has the different partition
(Motion vector of block A is MVa, block B is MVb, block C is MVc)

Prediction and different Motion Vector of current block (MVp and MVd) and can be calculated as:

$$MVp = Median(MVa, MVb, MVc)$$
$$MVd = MVr - MVp \qquad (2\text{-}1)$$

where MVr is the real motion vector of current block after motion estimation.

In previous standards, an area (a collection of macroblocks) which did not have any motion in current frame could be simply replaced by co-located macroblocks in the reconstructed reference frame(s). So skip mode in the prior standards is also called copy mode. In contrast to previous standards, in H.264/AVC there is predicted motion for skipped areas which alleviates the detrimental effects on the rate distortion performance in video sequences containing "global" motion. The motion vector of skip mode macroblocks in H.264/AVC is generated identically to the prediction motion vector for the 16x16 motion compensated macroblock mode.

In both special cases, a zero motion vector is used to for stationary background in current frame:

1) If the up or the left neighboring macroblock of current macroblock is not available.

2) If the prediction motion vector of current macroblock is zero and the latest frame is used as reference frame in motion compensation.

No motion information and no predictive error coding data are sent for skip mode macroblocks. Coding efficiency can be improved especially for slow motion video sequences.

H.2 Direct Mode

A.M. Tourapis in [33] proposed a new mode called "Direct Mode" (adopted by H.264/AVC) to exploit the temporal correlation by bi-directional prediction from both forward and backward reference pictures. This mode does not require any bits for coding the motion vectors but predictive error coding data needs to be sent.

Direct mode has two motion vectors (forward motion vector MV_0 and backward motion vector MV_1) which are generated from the motion vector MV_C of the co-located macroblock of the next reference picture RF_1. The predicted best match is calculated by the linear combination of two blocks determined by MV_0 and MV_1 pointing to two reference pictures (RF_0, RF_1). Figure 2-9 and equation 2-2 illustrate the generation of

motion vectors of direct mode.

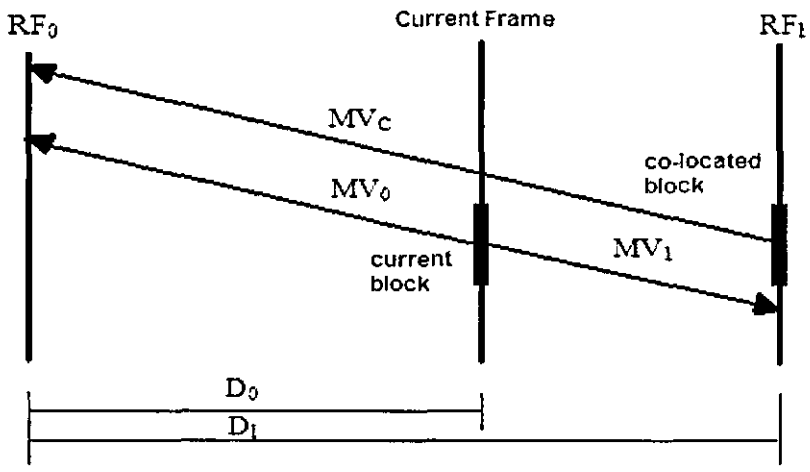


Figure 2-9 MV_0 and MV_1 of a Direct-mode

$$MV_0 = \frac{D_0}{D_1} MV_C$$
$$MV_1 = \frac{D_0 - D_1}{D_1} MV_C \qquad (2\text{-}2)$$

where D_1 is the temporal distance between the reference frames RF_0, RF_1, and D_0 is the distance between the current frame and reference frame RF_0.

I. Edge Extrapolation for Intra Coded Areas

This new feature is applied to the reconstructed version of the current picture for areas encoded in intra mode. It improves the quality of the predictive signal for motion compensation of subsequent pictures and it also allows prediction from neighboring inter coded areas in the reconstructed current picture that was not supported in H263+ or MPEG-4.

A prediction block is formed based on previously coded and reconstructed blocks and the difference between current block and the prediction is coded. There are 9 directional modes for luma samples when the macroblock is divided into 4x4 blocks (Intra 4x4 mode) and 4 directional modes are used when the macroblock is 16x16 (Intra 16x16 mode). The chroma samples can also be predicted in 4 different modes

similar with the Intra 16x16 case. In Intra 4x4 mode, each 4x4 luma samples are predicted from spatially neighbouring samples as illustrated figure 2-10 and 9 different directional ways of performing the prediction can be selected by the encoder as illustrated in figure 2-11 (eight directions of prediction modes and DC mode are plotted). The 4 directional prediction modes for Intra 16x16 and chroma modes are shown in figure 2-12. 8x8 luma intra prediction called Intra 8x8 can also be selected in H.264/AVC FRExt (Fidelity Range Extension) [34]. Intra 8x8 uses the same concepts as intra 4x4.

Q	A	B	C	D	E	F	G	H
I	a	b	c	d				
J	e	f	g	h				
K	i	j	k	l				
L	m	n	o	p				

Figure 2-10 A 4x4 Block and its Neighbour Samples
(Low case letters are current samples and capital letters are neighbour samples)

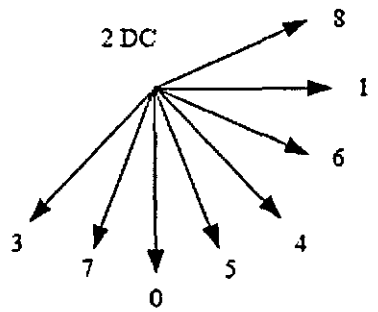


Figure 2-11 9 Modes of Direction for Intra 4x4

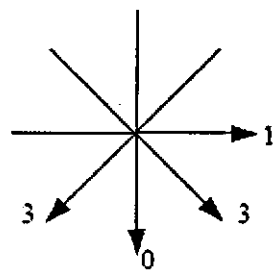


Figure 2-12 Directions of Intra 16x16

D. Marpe and V. George in [35] make an analysis of performance comparison between Motion-JPEG2000 [36] and H.264/AVC in pure intra coding mode. Experiments show that the H.264/AVC intra modes achieve similar quality performance compared with Motion-JPEG2000 in most case excluding very high bit-rates, but the computational complexity of H.264/AVC intra modes is much lower than the Motion-JPEG2000.

J. Hierarchical Block Size Transform Allowing Small Block Sizes

The smallest block size transform allowed in the standard is 4*4 (as opposed to the classic 8*8 size in older standards) and its distinct benefit is the ability to represent the signal in a more locally adaptive fashion where it is needed. Of course, transform block sizes up to 16*16 are also allowed for signals of sufficient correlation that can use longer basis functions (compared to the standard 8*8 DCT ones). The hierarchical order of the transforms [37] enables, among other benefits, the extension of a 4*4 to an 8*8 transform for low frequency chroma information and the use of a special type intra coding for extending the block size to 16*16 for low frequency luma information.

K. In Loop Deblocking Filter

For removing compression artifacts introduced by motion estimation and transform coding, a better designed adaptive deblocking filter [38] (as compared to H263+) is included in H.264/AVC inside the motion compensation loop. The filter is applied to the vertical and horizontal edges of 4x4 blocks in a macroblock. Figure 2-13 shows four samples on vertical and horizontal edges in neighbor bocks. The amount of filtering is measured by edge strength which is decided by the quantization parameter, modes of the neighbor blocks and the gradient of the samples across the edge. Two thresholds defined in H.264/AVC determine the decision on whether to perform filtering on the current boundary or not. The adaptive deblocking filter's aim is to smooth the blocking edges around the boundary of each macroblock without affecting the sharpness of the picture. This improvement in objective and subjective reconstructed signal quality has beneficial effects for subsequent motion compensations using the filtered picture as reference. The filter affects the subjective quality of decoded frames and improves coding efficiency.

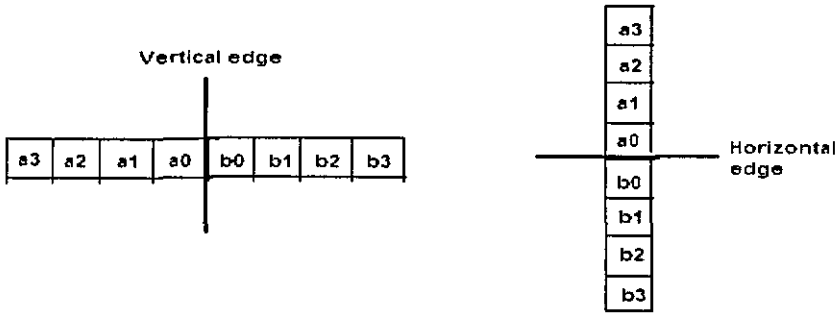


Figure 2-13 Samples on Vertical and Horizontal Edges

L. Exact Match Inverse Transform

The ability of H.264/AVC to utilise effectively integer transforms with performance very similar to non integer ones makes it the first video coding standard that can produce a completely "drift free" decoded representation. All previous standards (with the exception of an optional part of H263+) could not achieve this, since they were all using floating point transforms resulting in round off errors due to different machine representations.

M. Advanced Entropy Coding Techniques

H.264/AVC uses two very powerful entropy coding techniques namely Context Adaptive Variable Length Coding (CAVLC) and Context Adaptive Binary Arithmetic Coding (CABAC) [39] which are improved entropy coding mechanisms as compared to the ones mentioned in section 1.1.2-E. CAVLC exploits inter-symbol redundancy by switching between VLC tables for various syntax elements, depending on correlations of previous coded symbols. The advantage of CAVLC is lower computation compared with CABAC. CABAC achieves good compression performance through selecting probability models for each syntax element according the element's context, adapting probability estimations based on local statistics and using arithmetic coding. CABAC can increase coding efficiency by roughly 10% relative to CAVLC, but CABAC is significantly more computationally complex. Both these techniques capitalize on context adaptivity to improve rate distortion performance. The only previous standard that included optional

arithmetic coding was H.263.

N. Provisions for embedded processors

Compared to DCT, integer transforms which have only integer numbers ranging from -2 to 2 in the transform matrix are implemented in H.264/AVC standard. This allows computing the transform and the inverse transform with coefficients and scaling factors in 16-bit arithmetic using only low complex shift, add, and subtract operations. Therefore H.264/AVC is realizable on 16 bit processors, while all other standards required a minimum of 32 bit processors. This definitely makes the standard more appealing for embedded applications.

2.3 Complexity Analysis of H.264/AVC

However, the improvements in statistical/visual quality for the same bit rates as previous standards come with significant complexity increases. These new features not only increase the complexity of H.264/AVC encoder but also increase the complexity of H.264/AVC decoder. Variable block size motion estimation and compensation, hadamard transform, RDO mode decision, displacement vector resolution, multiple reference frames are main major H.264/AVC encoding tools which increase the complexity of H.264/AVC encoder. [40-42] present some analysis about the complexity increase of H.264/AVC video coding standard compared with other previous standards. The significant computational complexity makes it very difficult to use H.264/AVC video coding standard in real-time application systems. Reducing the complexity without degrading video quality becomes a critical problem.

In order to make us understand the complexity of H.264/AVC more clearly, an experiment of complexity analysis is performed here. The Intel® VTune™ Performance Analyzer7.0 is used in this work as the evaluation tool to evaluate the software performance and obtain the complexity profile of the H.264/AVC encoder. In this experiment, the Foreman sequence (100 frames, QCIF format, Baseline profile) is encoded on an Intel Pentium-4 3.09GHz PC with 768 MB memory under the Microsoft Windows XP. Fig. 2.14 shows the complexity proportion of different encoding modules

in the H.264 JM8.1 [43] reference encoder.

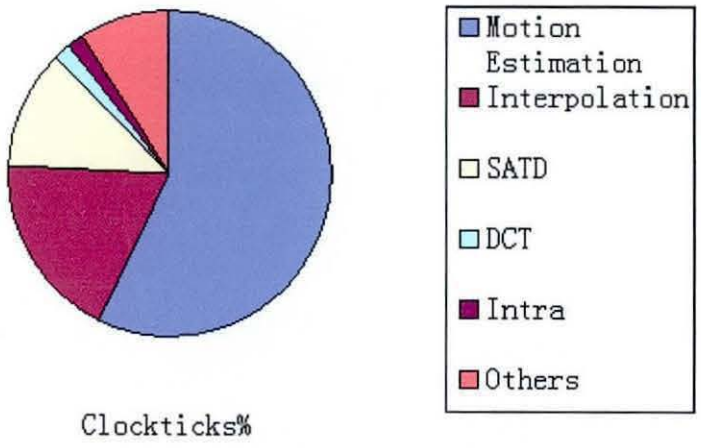


Figure 2-14 Complexity proportion of different encoding modules in H.264/AVC encoder by Intel® VTune™

According to Fig. 2-13, the most time-consuming modules of H.264/AVC encoder are Motion Estimation, Interpolation, SATD, and DCT which are all related with the RDO based the motion estimation and mode decision. Therefore a good fast mode prediction algorithm for H.264/AVC is a promising way to reduce the complexity of H.264/AVC encoder.

Chapter 3

Mode Decision in H.264/AVC: Preliminaries and Literature Review

3.1 Preliminaries

3.1.1 Preamble

Macroblocks in H.264/AVC standard have many mode candidates which can be chosen because of variable block size motion estimation and directional intra modes. Consequently, some criteria need to be used to decide which mode candidate is the best one for current macroblock. H.264/AVC standard adopts two mode decision schemes: high-complexity mode decision also known as RDO based mode decision and low-complexity mode decision. Compared with low-complexity mode decision, RDO based motion estimation and mode decision improves PSNR (up to 0.35 dB) and bit rate (up to 9% bit savings) in H.264/AVC standard and also comes with significant computational complexity. Because of the importance of mode decision in H.264/AVC standard, section 3.1 gives a brief overview of the mode decision mechanism.

3.1.2 Low-complexity Mode Decision

In low-complexity mode decision, the cost of each prediction mode is calculated using some biased of Sum of Absolute Difference (SAD) or the Sum Absolute Transformed Difference (SATD) of the prediction errors. The flow chart is shown in

figure 3.1.

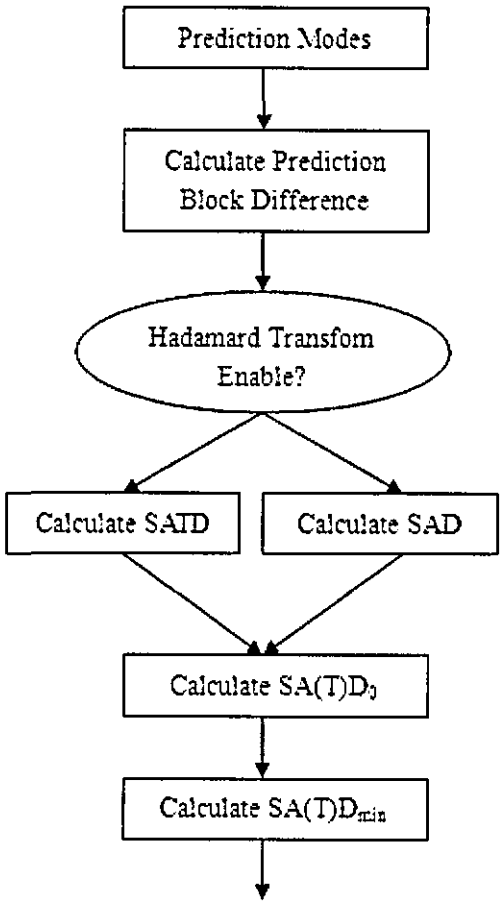


Figure 3.1 Flow Chart of Low Complexity Mode Decision

Firstly, the box of calculate $SA(T)D_0$ in the diagram will be explained here. The SA(T)D to be minimised is given a 'bias' value $SA(T)D_0$ initially in order to favour prediction modes that need few bits to be signaled. This bias is basically the multiplication of a parameter representing bit usage for reference frame and different motion vectors and a parameter $QP_0(QP)$ which is a table lookup with input quantization parameter (QP).

The calculation of $SA(T)D_0$ at each mode should be presented as follows.

- Forward prediction mode:

$$SA(T)D_0 = QP_0(QP) \times (2 \times code\_number\_of\_ref\_idx\_fwd + Bits\_to\_code\_MVDFW) \quad (3\text{-}1)$$

- Backward prediction mode:

$$SA(T)D_0 = QP_0(QP) \times Bits\_to\_code\_MVDBW \quad (3\text{-}2)$$

- Bi-directional prediction mode:

$$SA(T)D_0 = QP_0(QP) \times (2 \times code\_number\_of\_ref\_idx\_fwd + Bits\_to\_code\_forward\_Blk\_size + Bits\_to\_code\_backward\_Blk\_size + Bits\_to\_code\_MVDFW + Bits\_to\_code\_MVDBW) \quad (3\text{-}3)$$

- Direct prediction mode:

$$SA(T)D_0 = -16 \times QP_0(QP) \quad (3\text{-}4)$$

- Intra 4x4 mode:

$$SA(T)D_0 = 24 \times QP0(QP) \quad (3\text{-}5)$$

- Intra 16x16 mode:

$$SA(T)D_0 = 0 \quad (3\text{-}6)$$

The Prediction error block is the difference between the original and prediction blocks.

$$Diff(i,j) = Original(i,j) - Prediction(i,j) \quad (3\text{-}7)$$

Regarding the box calculate SATD in the diagram, a two dimensional transform is performed in the decision loop for selecting predicted modes for current block when the transform option is enabled. To simplify implementation, the hadamard transform is

chosen. A 4x4 hadamard transform matrix is shown below:

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

This transform is performed horizontally and vertically and results in DiffT(i, j) which is called hadamard transform of difference between current and predictive block. The SATD for the block and for the present prediction mode is produced as shown in the equation below.

$$SATD = (\sum_{i,j} |DiffT(i,j)|)/2 \quad (3\text{-}8)$$

Finally, the prediction mode that achieves the minimum value is chosen according to:

$$SA(T)D_{min} = \min(SA(T)D + SA(T)D_0) \quad (3\text{-}9)$$

3.1.3 High-complexity Mode Decision

A. Introduction to Rate Distortion Theory

The starting point of classical Rate distortion theory was found by Claude Shannon in his seminal work [44, 45] in the information theory community. Rate distortion theory [46] has been actively studied for the last 50 years, and [47-49] are two main contributions which used Rate distortion theory into the field of video compression.

Rate distortion theory is a major branch of information theory; it addresses the problem of determining the minimal amount of entropy (or information) R that should be communicated over a channel, so that the source (input signal) can be reconstructed at the receiver (output signal) with given distortion D. Figure 3-2 shows the relation between the entropy R and the distortion D.

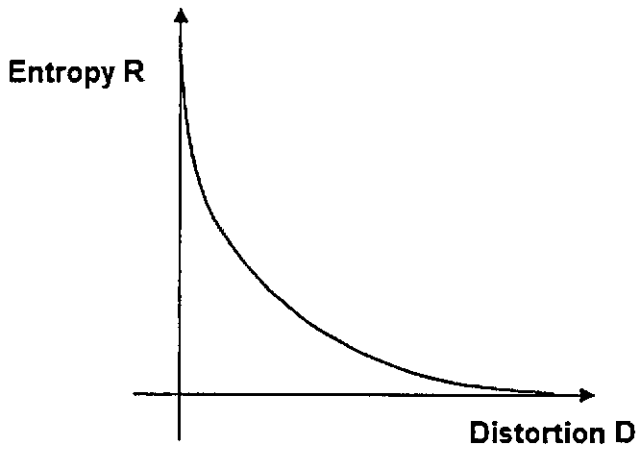


Figure 3-2 - Relation of Entropy R and Distortion D

The rate-distortion theory in video compression field can be formulated as an R-D optimized mode decision problem which is to find the mode that minimizes the quantization distortion D_q for a given MB, subject to a constraint on the number of bits R_c used. This constrained problem can be obtained by solving as a minimisation problem below.

$$\min_{M_i^n} D_q(M_i^n) \qquad \text{subject to } R(M_i^n) < R_c \tag{3-10}$$

Where $D_q(M_i^n)$ and $R(M_i^n)$ denote the quantization distortion and the number of bits used, respectively, for macroblock with a particular mode M_i^n. The Lagrangian multiplier optimization method [50-52] is well known for solving this kind of constrained minimization problems. Instead of the cost function $D_q(M_i^n)$, with constrained $R(M_i^n)$, or $R(M_i^n)$, with constrained $D_q(M_i^n)$, an unconstrained Lagrangian cost function can be used to solve the problem:

$$J = D_q(M_i^n) + \lambda R(M_i^n) \tag{3-11}$$

Where Lagrangian multiplier λ is a non-negative real number.

B. RDO Based Mode Decision for H.264/AVC

In the high complexity mode of H.264/AVC standard, the macroblock mode decision is done by minimising the Lagrangian functional:

$$J(s,c,MODE \mid QP,\lambda_{MODE}) = SSD(s,c,MODE \mid QP) + \lambda_{MODE} * R(s,c,MODE \mid QP) \quad (3-12)$$

In the above equation, J denotes the cost function and is dependent on s (the original signal macroblock), c (the reconstructed signal macroblock) and MODE (selected from a set of modes as explained later). J is found given QP (the macroblock quantisation step size) and λ_{MODE} (the Lagrange multiplier for mode decision). SSD is the sum of the squared differences between the original macroblock and its reconstruction with QP and it also depends on the original and reconstructed macroblock, as well as the mode decision (MODE). It is given by the following equation:

$$\begin{aligned} SSD(s,c,MODE \mid QP) = & \sum_{x=1}^{16}\sum_{y=1}^{16} s_Y[x,y] - c_Y[x,y,MODE \mid QP])^2 + \\ & + \sum_{x=1}^{8}\sum_{y=1}^{8} (s_U[x,y] - c_U[x,y,MODE \mid QP])^2 + \\ & + \sum_{x=1}^{8}\sum_{y=1}^{8} (s_Y[x,y] - c_Y[x,y,MODE \mid QP])^2 \end{aligned} \quad (3-13)$$

where $c_Y[x,y,MODE \mid QP]$ and $s_Y[x,y]$ represent the reconstructed and original luminance values and c_U, c_V, s_U, s_V the corresponding chrominance values.

The Lagrange multiplier λ_{MODE} depends on the slice type (including SP slices that are similar to P slices but can afford identical reconstruction even when predicted from different reference pictures), the number of slices in the sequence and on the quantisation step size per macroblock according to the following relations:

For a single B slice in a sequence:

If the macroblock belongs to that slice: $\lambda_{MODE} = 0.68 * 2^{QP/3} * MAX(2, MIN(4, QP/6))$

If the macroblock belongs to an SPslice:

$\lambda_{MODE} = 0.68 * 2^{QP/3} * MAX(1.4, MIN(3, QP/12))$

If the macroblock belongs to an Intra or P slice: $\lambda_{MODE} = 0.68 * 2^{QP/3}$

For more than one B slices in the sequence:

If the macroblock belongs to a B slice: $\lambda_{MODE} = 0.85 * 2^{QP/3} * 4$

If the macroblock belongs to an SPslice:

$\lambda_{MODE} = 0.85 * 2^{QP/3} * MAX(1.4, MIN(3, QP/12))$

If the macroblock belongs to an Intra or P slice: $\lambda_{MODE} = 0.85 * 2^{QP/3}$

Finally, the rate $R(s,c,MODE \mid QP)$ depends on the original and reconstructed macroblock with quantisation parameter QP, as well as the chosen MODE, and reflects the number of bits produced for header(s) (including MODE indicators), motion vector(s) and coefficients.

In equation 3-12, MODE is chosen from the set of potential prediction modes as follows:

For Intra slices: $MODE \in \{INTRA4*4, INTRA16*16\}$ (3-14)

For P slices: {single reference forward or backward prediction}

$$MODE \in \{INTRA4*4, INTRA16*16, SKIP, MODE\_16*16, MODE\_16*8, MODE\_8*16, MODE\_8*8\} \quad (3-15)$$

For B slices: {bi-directionally predicted slices}

$$MODE \in \{INTRA4*4, INTRA16*16, DIRECT, MODE\_16*16, MODE\_16*8, MODE\_8*16, MODE\_8*8\} \quad (3-16)$$

DIRECT mode is particular to the bi-directionally predicted macroblocks in B slices, while SKIP mode implies that no motion or residual information will be encoded (only the MODE indicator is actually transmitted).

In the above mode sets, any mode with the prefix INTRA will result in encoding the signal rather than its residual, while any mode with the prefix MODE_ refers to inter modes. Furthermore, when MODE is equal to INTRA4*4 or INTRA16*16, the best intra mode for each case is chosen through evaluation of the functional of equation 3-12 with mode choices from the following sets:

$$INTRA4*4 \in \{DC, HORIZONTAL, VERTICAL, DIAGONAL\_DOWN\_RIGHT, DIAGONAL\_DOWN\_LEFT, VERTICAL\_LEFT, VERTICAL\_RIGHT, HORIZONTAL\_UP, HORIZONTAL\_DOWN\} \quad (3-17)$$

$$INTRA16*16 \in \{DC, HORIZONTAL, VERTICAL, PLANE\} \quad (3-18)$$

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

$$MODE\_8*8 \in \{INTER\_8*8, INTER\_8*4, INTER\_4*8, INTER\_4*4\} \quad (3-19)$$

Any mode with the prefix MODE_ in equations 3-15, 3-16, 3-19 assumes that the best motion vector is known for this mode and implies functional minimisations for each candidate motion vector $m = (m_x, m_y)$ inside the search window of the form:

$$J(m, \lambda_{MOTION}) = D_{DFD}(s, c(m)) + \lambda_{MOTION} * R(m-p) \quad (3-20)$$

where λ_{MOTION} is the Lagrange multiplier for motion estimation, $p=(p_x, p_y)$ is a predicted motion vector and $R(m-p)$ is the number of bits for encoding motion residuals only. The rate term is computed from a look-up table, while the distortion term depends on the original signal and the reconstructed best match that in turn depends on the candidate motion vector. The expression for the distortion is given by

$$D_{DFD}(s, c(m)) = \sum_{x=1}^{16}\sum_{y=1}^{16} s_Y[x, y] - c_Y[x+m_x, y+m_y])^q \quad (3-21)$$

where s_y, c_y are the original and reconstructed signal blocks of the luma component and q is either 1 or 2, denoting the Sum of the Absolute Differences after Hadamard transform (SAD(T)D) or the Sum of the Square Differences respectively. It has to be noted that the choice of λ_{MOTION} in equation 3-20 is affected by the choice of the distortion metric and that λ_{MOTION} and λ_{MODE} are related through the expression:

$$\lambda_{MOTION} = \sqrt{\lambda_{MODE}} \quad (3-22)$$

Once the best 8*8, INTRA4*4, INTRA16*16 modes are found, the minimal cost for the macroblock is evaluated by looping through the different mode possibilities (equations 3-15, 3-16).

A straightforward mode decision complexity assessment for a 16*16 macroblock (luma component only) reveals that we need 144 cost evaluations for the best INTRA4*4 mode. Adding 4 more evaluations for the INTRA16*16 case, 16 more for the best 8*8 inter mode and 7 more for selecting the minimal cost among all modes results in 148 evaluations for macroblocks in Intra slices and 171 evaluations for macroblocks in P or B slices. Coupled with similar cost evaluations for the chroma components and the fact that our complexity assessment did not consider evaluations for the best motion vector that depend on the size of the search window and on the sub-pixel accuracy (equation 3-21), clearly shows that the mode decision process is computationally intensive.

The average execution time of low-complexity mode decision is only 7% of that of high-complexity mode decision. But low-complexity mode decision loses an average of 0.48dB in PSNR compared to high-complexity mode decision. In the following thesis, unless we mention low-complexity mode decision explicitly, mode decision means high-complexity mode decision.

3.2 Literature Review

3.2.1 Preamble

Section 3.2 is a literature review of existing fast mode prediction of H.264/AVC standard. In this chapter, four categories of fast mode prediction—skip and direct mode prediction, fast inter mode prediction, fast intra mode prediction and fast intra/inter mode decision—will be classified. Different mode prediction approaches in each category will be discussed and compared. The comparison will enable us to better understand the assumptions, advantages and limitations of these approaches.

3.2.2 Skip Mode Detection

In some prior standards, for example H.263+, a "skipped" macroblock is defined as a macroblock with zero Motion Vector and zero Quantized Transform Coefficients. A significant proportion of MBs are skipped (not coded), particularly in low-motion sequences and/or at higher quantizer step sizes (and hence lower bitrates). Y. Zhao and I. E. G. Richardson in [53] proposed a skip mode detection technique for this kind of "skipped" macroblock. In [53] SAD_0 (SAD (Sum of Absolute Difference) between the current macroblock and the co-located macroblock) is used to judge "skipped" macroblocks. If SAD_0 over quantization parameter is smaller than an adaptive threshold, current Macroblock is "skipped" macroblock. The difference of "skipped" macroblocks between prior standards and H.264/AVC is that: in prior standards, a "skipped" area of a predictively-coded picture could not move, while in H.264/AVC a predefined predicted motion which is derived directly from previously encoded information is used in skipped areas. Apparently, skip mode detection in [53] does not consider about predicted motion for skipped mode in H.264/AVC, so it is not suitable for H.264/AVC standard and new skip mode schemes need to be found.

C.S. Kannangara in [54] proposed a new skip mode detection for low complexity mode decision in H.264/AVC standard. This early prediction is made by estimating a Lagrangian rate-distortion cost function which incorporates an adaptive model for the Lagrange multiplier parameter based on local sequence statistics. None of Motion estimation needs to be done. 19%-67% computational saving can be achieved without significant loss of rate-distortion performance. The experiments prove that [54] achieve very similar Rate distortion performance compared with original low complexity mode decision in H.264/AVC, but [54] does not work well in RD based mode decision scheme.

Previous work on SKIP mode prediction has resulted in contributions to the high complexity mode decision of H.264/AVC standard both for the P and B slices and for the full and fast motion estimation algorithms (FS and FME) [55-57]. According to this work, a macroblock can be predicted as having SKIP mode in the baseline profile (P slices only) [55], when the following set of four conditions is satisfied:

1) The best motion compensation block size for this macroblock is 16x16 (inter MODE_16*16)

2) The best reference slice is the previous slice

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

4) The transform coefficients of the 16x16 block size are all quantised to zero.

For the B slices of the main profile [56], a macroblock can be predicted as having SKIP mode when the following set of two conditions is satisfied:

1') The reference slices and the motion vectors are the same as the ones decided under the direct mode.

2') The transform coefficients of the 8x8 sub-blocks of this macroblock are all quantised to zero.

The advantage and limitation of [55] and [56] will be mentioned in chapter 4.

3.2.3 Fast Inter Mode Prediction

A. Splitting and Merging Mode Decision

Fixed-size block-based Motion Estimation (ME) has been widely adopted by international standards such as MPEG-1, 2, 4 and H.261/H.263. Recently, variable-sizes block-matching (VSBM) has been adopted by ITU-T H.264 [58]. Although VSBM can achieve significant video quality improvement, it requires high computational complexity for determining the best partition of a macroblock for ME. The exhaustive–search method needs to perform motion estimation with all block–sizes defined in the standard and choose the best one which minimizes a cost function (such as Rate Distortion Lagrangian function).

To reduce the complexity of VSBM, fast mode decision before motion estimation

process should be investigated. M. Chan in [59] proposed a "top-down" approach in which initially large blocks are matched. If the SAD (Sum of Absolute Difference) of the best match is above a predefined threshold, then the block is further split into small blocks. This process is repeated until the maximum number of blocks, or locally minimum errors, are obtained. In the "top-down" algorithm, if one larger block needs to be splitted, one should perform motion estimation for those smaller blocks to find the best matches of them to achieve higher PSNR. So there is a trade-off between complexity and rate distortion performance. To overcome this disadvantage, Injong Rhee in [60] proposed a "bottom-up" VSBM algorithm which relies more on local motion information. A set of "candidate" motion vectors of each small block is first obtained during full-search motion estimation, whose matching error is less than a prescribed threshold. Neighboring blocks are then merged if they have at least on vector in common. This "bottom-up" VSBM algorithm does not need to do the motion estimation again even if we find that some small block should be merged. But the choice of threshold becomes a key problem. Too large or too small thresholds all result in bad rate distortion performance. Yu-Kuang in [61] proposed an adaptive threshold for the merging algorithm. SAD and Quantizer Parameter are both taken into account in order to decide the current threshold. From the results reported, we can say that in order to keep the quality in a good level, the algorithm uses much higher bit rates than the H.264/AVC standard.

Experiments in those papers using splitting and merging mode decision technique show that either the quality loss is too much at the same bitrate or the bitrate increase is too large at the same quality compared with the original video coding standards.

B. Fast Inter Mode Decision

In H.264, there are altogether 7 different block sizes (16x16, 16x8, 8x16, 8x8, 8x4, 4x8, and 4x4 blocks) that are used per macroblock. Motion estimation and Rate-Distortion Optimization (RDO) are performed using all the block sizes in order to find the best in RD terms, resulting in heavy computational load at the encoder. If we can predict the optimal block partition earlier, time savings will be significant.

X. Jing, L. Chau in [62] proposed a fast inter mode decision method which only

depends on the MAD between the current frame and previous frame. The main idea is to use large block types for smooth areas and small blocks for areas containing complex motions. If the MAD of current macroblock and co-located macroblock is smaller than the weighted mean absolute frame difference (MAFD) between current frame and previous frame, large block types (16x16, 16x8, 8x16) are chosen. Otherwise all block types are chosen. The disadvantage of this algorithm is that the weighting which is used in fast mode decision should be set based on quantisation parameters (QP). Different QPs need different weights, which makes this algorithm is not very practical in the real applications. This proposed algorithm can obtain up to 48% of computations saving with similar rate distortion performance with H.264/AVC standard.

A. Chang in [63] proposed another algorithm which still uses SAD information to predict inter modes. If the texture undergoes an integer-pixel translational motion, the texture will look exactly the same in the two consecutive frames and it can be predicted perfectly by integer-pixel motion estimation. If the edges of the texture have a half-pixel or quarter-pixel offset, the edges may be blurred, sub-pixel motion estimation will be important. SAD1 (SAD of mode 16x16) is calculated after motion estimation on mode16x16. If SAD1 is smaller than an adaptive threshold based on the average of SAD1s of macroblocks which choose mode 16x16 to be the best mode, the current macroblock satisfies with the condition of integer-pixel translational motion and mode 16x16 is the best mode. Otherwise, current macroblock may have a half-pixel or quarter-pixel offset. Texture analysis and segmentation will be done on the best match of mode 16x16 to choose mode 16x8 or 8x16 to be the best mode. However [63] only consider three coding modes which are 16x16, 16x8 and 8x16 inter coding modes. With the consideration of only three modes, 40.73% of computational cost reduction can be achieved in [63] with similar visual quality and bitrate.

Motion vector information has been used in [64] to predict inter modes. This algorithm is based on the principle of 3D recursive search algorithm [65] which provides a fast convergent and highly accurate motion vector prediction, taking into account the modes calculated in the previous and current frames and their motion vector cost. The motion vector cost is calculated using the Lagrange multiplier and motion vectors.

Candidate modes are chosen from the left, top, and top-left of the current macroblock in current frame and the macroblock which is two rows down and one column right of the current macroblock in the previous frame. The mode with the minimal motion vector cost is chosen to be the best mode for the current macroblock. Using this algorithm, a maximum increase of about 15% bitrates achieve at the same quality compared with standard. As can be seen, increase of bitrates is too much at the same quality and will effect negatively the Rate Distortion performance.

A.C. Yu in [66] proposed an inter mode prediction algorithm relying on two factors, complexity of macroblock and mode information from previous frame. DCT coefficients are calculated after motion estimation. Based on the analysis of DCT coefficients, macroblocks can be classified into homogeneous macroblocks, medium-detailed macroblocks or high-detailed macroblocks. Each category includes a set of mode candidates. Because of the high temporal correlation between the modes of the current macroblock and co-located macroblocks in previous frame, the mode information in the previous encoded frame is used to modify the mode groups in the three categories and improve the accuracy of mode prediction. The proposed algorithm can save up to 31% of the encoding time as compared with H.264/AVC standard without sacrificing both quality and bit rate efficiency too much.

K.P.Lim in [67] proposed a new algorithm called homogeneous regions detection to classify the inter modes into groups. It is observed that in non-deforming, smoothly moving video sequences, the various parts of the video objects move together. The so called "spatial smoothness of motion" has been commonly used in computing motion fields [68] [69]. One of the main reasons for using variable block sizes in H.264/AVC is to represent motion of video objects more adequately. Since homogeneous regions tend to move together, homogeneous blocks in the frame should have similar motion and should not be further split into smaller blocks. So some macroblocks which are homogeneous could belong to a part (16x16, 16x8, 8x16) of all the modes, and do not need to do the motion estimation and mode decision for all modes. According to the work in [67], a region is homogeneous if the texture in the region has very similar pixel values and such a texture can be represented by blocks of bigger size. In [67], the authors use the edge

map computed in the fast intra mode decision technique of [70] to judge which macroblocks belong to homogeneous regions. If the current macroblock is homogeneous, its mode should belong to the set {16x16, 16x8, 8x16}. Otherwise we should test all the modes. The results in [67] show that this algorithm can speed up 30% of the encoding process with maximum 0.08dB of PSNR decrease and maximum 1.44% of bit rate increase. There are some other contributions which use different techniques to divide modes into groups and we refer the reader to [61, 77, 95].

D. Zhu in [71] uses a 7-tap filter on horizontal and vertical directions of the original and reference images respectively to get down-sampled half resolution small images. The prediction mode selection method of [67] is performed to get a set of prediction mode candidates in small images. This set of mode candidates will be mapped to a set of mode candidates for current macroblock in original image. Because motion estimation is performed in small images, a small set of mode candidates will be chosen and thus a lot of time saving can be achieved.. The experimental result shows that this algorithm can reduce nearly 50% of encoding time with PSNR reduction about 0.2dB.

M. Yang and W. Wang in [72] proposed a fast inter mode decision based on fuzzy classification theory [73] which exploits spatial correlation. Each macroblock is classified into two categories, complex motion macroblock and simple motion macroblock. Different inter mode candidates are assigned to the two categories, based on early termination techniques.

SATD-based costs in Low-complexity mode decision (section 3.1.2) have been used in [74, 75] for fast high-complexity (RD-based) mode decision. There are two mode decision schemes in H.264/AVC, low-complexity mode decision using the sum of SATD, estimated rates and quantization parameters for mode selection and high-complexity using RDO. Compared to the latter one, low complexity mode decision results in degradation of video quality at the same bit rate. However experiments proved that a MB mode selected by low complexity mode decision can be a good indicator that the candidate mode is highly probable that it also can be selected by RD based mode decision. This means there exists a strong relationship between the costs of low complexity mode

decision and the costs of RD based mode decision. In these algorithms, three most probable modes with lowest costs in low-complexity mode decision have been chosen for high complexity mode decision. The disadvantage of this idea is that mode candidates are known only after all motion estimation has been performed, thus only part of mode decision process can be saved timewise. About 80% of execution time of RD based mode selection can be saved while average PSNR loss is 0.07dB.

A fast inter mode decision algorithm is proposed in [76] by exploiting the correlation of mode J costs. The basic idea of this algorithm is that if the cost of larger block-size modes is higher than the cost of the current block-size mode, then the best mode of current macroblock cannot be of an even larger block-size. Meanwhile, if the cost of a smaller block-size mode is higher than that of current block-size mode, then best mode of current macroblock cannot be even smaller. A similar idea has been used in [77] which is based on the monotonicity of the error surface as another way to group the modes. The error surface is built initially by 3 modes: 16x16, 8x8 (the entire macroblock is examined using only 8x8 partitions which means four 8x8 sub-blocks for this macroblock), 4x4 (the entire macroblock is examined using only 4x4 partitions which means sixteen 4x4 sub_blocks for this macroblock). If the error surface is monotonic, that is if J(16x16) < J(8x8) < J(4x4) or J(16x16) > J(8x8) > J(4x4) where J denotes a cost function, only modes (block sizes) between the best two modes are tested. If not, all other modes need to be tested. The disadvantage of this idea is that the order of motion estimation for different partition block size needs to be change which will introduce complexity and will affect negatively the Rate distortion performance.

3.2.4 Fast Intra Mode Prediction

Intra mode prediction is applied to the reconstructed version of the current picture for areas encoded in intra mode. It improves the quality of the prediction signal for motion compensation of subsequent pictures and it also allows prediction from neighboring inter coded areas in the reconstructed current picture. This feature was not supported in H263+ or MPEG-4.

If a block or macroblock is encoded in intra mode, a prediction block is formed based on previously encoded and reconstructed (but un-filtered) blocks. This predictive block is subtracted from the current block prior to encoding. For the luminance (luma) samples, this block may be formed for each 4x4 sub-block or for a 16x16 macroblock. There are a total of 9 prediction modes for each 4x4 luma block and 4 modes for a 16x16 luma block. (The video format we talk about below is 4:2:0.) Each 8x8 chroma component of a macroblock is predicted from chroma samples in the above and/or left chroma blocks that have previously been encoded and reconstructed. The 4 chroma prediction intra modes are very similar to the 16x16 luma prediction modes. The same prediction mode is always applied to both chroma blocks. Furthermore, if any of the 16x16 macroblocks in the luma component are coded in Intra mode, both chroma blocks are Intra coded but the prediction intra mode of the chroma is selected independently of the prediction type for the luma.

That means for one macroblock, there are 4 x (9 x 16 + 4) =592 combinations of intra modes we should consider to find the best intra mode for this macroblock. As a result, the complexity and computation load of the encoder are extremely high. If we can find some techniques to reduce the amount of RDO calculations for intra prediction, speeding up of intra coding will be realizable.

B. Meng [78] proposed a fast intra-prediction algorithm for intra4x4 blocks using low-complexity cost. Pixels in 4x4 blocks are categorized into 4 groups, and each group is a "down-sampled" version of the original block. The best prediction mode can be chosen with minimal complexity cost using SAD and quantisation parameter by checking some of these groups. Because of the correlation of directions of intra modes shown in figure 2-10, the best prediction mode's two neighbouring directional modes are also chosen as candidate modes, so in the end 3 intra4x4 prediction modes are chosen as candidate modes. To improve the speed of intra mode prediction, thresholds for early termination and neighbouring blocks' mode information are also used. Computation can be saved by not only examining a small set of intra4x4 mode candidates but also by using fewer pixels which are "down-sampled" from the original block for calculation. In order to achieve big time saving and good Rate distortion performance, different thresholds for

different video sequences based on different quantization parameters need to be set before encoding. This kind of setting thresholds cannot work very well in one pass encoding with video of unknown contents.

Y. Zhang's fast intra 4x4 mode prediction algorithm [79] is based on two observations. One is that the best intra 4x4 mode in 4x4 blocks is highly likely to be in the dominant direction of the local edges of those 4x4 blocks. The other is that authors observe DC mode has higher possibility to be the best mode than other intra 4x4 modes. A 4x4 block is divided into four 2x2 blocks and feature analysis is done to find the local edge direction. The local edge direction could be classified into 7 types, no obvious edge, vertical edge, horizontal edge, diagonal down/left edge, diagonal down/right edge, vertical-dominant edge and horizontal-dominant edge. A set of modes based on local edge direction information plus the DC mode are chosen to be the set of candidate modes. However there are two problems in the above algorithm which result in increase of bitrates and loss of PSNR. Firstly the assumption of the best mode in the dominant edge direction is not always true and secondly analysis of local edge information extraction based on the 2x2 blocks' intensity can not be very accurate. 40% to 70% of computational complexity can be saved with less than 5.5% of bit rate increase and not more than 0.05dB PSNR sacrificed.

Feng Pan, Xiao Lin et. al in [70] proposed a novel fast intra mode decision algorithm for H.264. This algorithm is based on local edge directional information in order to reduce the amount of calculations in intra prediction. Firstly, the Sobel edge operators [80] are applied to the current frame to generate the edge map. Then an edge direction histogram is calculated from all the pixels in the block by summing up the amplitudes of those pixels with similar directions in the block. The cell with the maximum amplitude indicates that there is a strong edge presence in that direction, and thus could be used as the direction for the best prediction mode. Finally with the use of edge direction histogram derived from the edge map of the picture, only a small number of the most likely intra prediction modes are chosen for RDO calculation. The drawbacks are similar with [79]. Using [70], average of 60% time saving can be achieved with average 5.9% of bitrate increase and negligible loss of PSNR.

In Pan's algorithm [70], an edge map needs to be produced for the whole picture which needs some computation. F. Fu in [81] proposed a faster algorithm which performs only partial edge detection. Based on the experiment, F. Fu observed that mode decision normally depends on the edge information between the left, top blocks and current block. So instead of doing edge detection in all picture, F. Fu only compute the edge information in the boundaries of blocks. The set of candidate modes are chosen based on Pan's intra candidate mode selection but using partial edge information. A most probable intra mode type for current macroblock can also be obtained based on intra mode types and costs of the left and top macroblocks of current macroblock, which can be an early termination. If the most probable intra mode type is intra4x4, and the cost of best intra4x4 mode is very large, Intra16x16 will not be chosen. If the most probable intra mode type is intra16x16 and the cost of best intra 16x16 is very small, Intra4x4 do not need to be examined. Compared with [70], [81] can reduce computation time by about 2 times but achieve a little worse rate distortion performance.

A fast intra16x16/intra4x4 selection algorithm is proposed in [82] using macroblock properties. The main idea is that intra16x16 modes are more suitable for predicting smooth areas and intra4x4 modes can achieve good prediction in regions with significant detail. Based on this idea, pixel level analysis based on thresholds is performed to divide macroblocks into two classes, smooth macroblocks and non-smooth macroblocks. Intra16x16 modes are chosen for smooth macroblocks and intra4x4 modes are chosen for non-smooth macroblocks. The proposed algorithm can achieve 10%-40% computation reduction with similar PSNR and bitrate performance compared with the H.264/AVC standard.

Changsung Kim in [83-85] proposed a new fast H.264/AVC intra prediction mode selection scheme. The proposed algorithm adopts a multi-stage mode decision process which uses a spatial domain feature (SAD (Sum of Absolute Difference)) and a transform domain feature (SATD (sum of the absolute transformed differences)) together to remove unlikely intra modes. In the final step of this scheme, A new Rate Distortion model is used to find the best intra mode from the set of intra modes when the quantization parameter (QP) is bigger than 16 due to the limitation of the accuracy of the R-D model.

Because the R-D model predicts the rate and distortion instead of getting the real rate and distortion, some computation can be further saved in the mode decision step. When QP is smaller than 16, RD-based mode decision will be used. Experiments show the reduction of computational complexity of intra RDO mode decision is up to 90% with little quality degradation at the same bitrates compared with the H.264/AVC standard..

C. Tseng in [86] introduced an improved cost function using absolute integer transform difference (SAITD) instead of SATD to achieve better rate distortion performance in the process for intra 4x4 mode decision in Low-complexity mode decision (section 3.1.2). In that work, a suite of fast transform algorithms for SATD and SAITD are also developed. The experimental results presented in [86] show that SAITD can indeed be a good indicator for fast intra mode prediction in RD based mode decision.

C. Cheng in [87] assumed the costs of intra4x4 modes are monotonic and proposed a three-step fast intra mode prediction algorithm. Motivated by the highly correlation of RD cost between different prediction direction, in each step of the algorithm, the neighborhood direction around the minimum one is explored efficiently and other unlikely intra modes can be skipped. After three steps, only 6 intra 4x4 prediction modes need to be examined instead of 9 intra4x4 modes in the H.264/AVC standard. Experiments show that this algorithm can obtain about 31% of average time saving intra RDO mode decision with similar PSNR quality and about 1% of bit rate increase compared with the H.264/AVC standard.

3.2.5 Fast Intra/Inter Mode Selection

Table 3-1 shows the ratio of Intra macroblocks selected in P and B slices in various video sequences at Qp=28 for Baseline and main profiles. As can be seen from the table, in real sequences, Intra macroblocks are selected very rarely in inter slices: the average ratio of Intra macroblocks in inter slices is about 1.40% (maximum 2.45% in container and minimum 0.33% in foreman) in Baseline profile and about 0.91% (maximum 2.13% in stefan and minimum 0.03% in mobile) in Main profile. From the numbers we can say if we can make a good decision of Intra/Inter Mode Selection before RDO mode decision,

RDO Intra mode decisions can be avoided for those macroblocks in which inter modes are the best ones and computational complexity of RDO mode decision can be reduced.

Profiles	News	Silent	Container	Foreman	Mobile	Stefan	Tempete
Baseline	0.75%	1.46%	0.33%	0.90%	2.07%	2.45%	1.87%
Main	0.69%	1.29%	0.49%	0.98%	0.03%	2.13%	0.74%

Table 3-1 The Ratio of Intra Macroblocks at Qp=28

Chen *et al.* [88] proposed a fast intra/inter mode selection scheme using the motion information (SAD) and the number of bits used to code the motion information for the MPEG-4 standard [30]. To facilitate video transmission over networks, Turaga and Chen [89] developed a classification-based mode decision scheme using the maximum likelihood (ML) criterion. However, the two schemes are not suitable for the intra/inter mode selection in H.264/AVC since the selected features are too simple to provide accurate mode prediction.

A feature-based fast intra/inter mode decision method is proposed by Changsung Kim in [90] to reduce the encoder complexity of the H.264 video coding standard. Firstly, the extracted features are the spatial domain correlation using SATD of the intra prediction residual, the temporal domain correlation using SATD of the difference between the current macroblock and the best matched one in reference frame, and the motion vector length. These three features are extracted from the current macroblock to form a 3D feature vector. Secondly based on the location of feature vector in the feature space, the video slices are partitioned into three regions: risk-free, risk-tolerable and risk-intolerable regions. Based on the small cells into which the risk region is quantized, the risk minimizing mode could be found. Finally, If the feature vector lies in the risk-free region, the mode decision is made based on simple feature comparison. If the feature vector lies in the risk-tolerable region, the risk minimizing mode is selected. If the feature vector is in the risk-intolerable region, a full RD based mode decision process is performed. Experiments show about 19-25% of total encoding time of H.264/AVC

standard can be saved at the expense of 4.1% average rate increase and 0.27% average distortion loss.

Finally, the selective intra mode decision proposed in [55] gives us a new technique with which we can decide whether the intra modes for the current macroblock should be checked or not. This technique uses *the average boundary error (ABE)* between the pixels on the boundary of the current and its adjacent encoded blocks under the best inter mode as an indicator of the degree of *spatial correlation* and *the average rate (AR),* i.e., the average number of bits consumed to encode the motion-compensated residual data under the best inter mode as an indicator of degree of *temporal* correlation. Then, we compare the average rate for the best inter mode and the average boundary error for the current block. If AR <k*ABE which k is a certain user-defined positive number, then we do not need to consider the intra mode for the current macroblock, otherwise we should consider it. Experiments show that about 20% of average time saving can be achieved in this algorithm at the expense of 0.036dB PSNR reduction and 0.7% bitrates increase.

Chapter 4

Fast Inter Mode Prediction for P Slices

4.1 Preamble

We propose an inter mode decision scheme [91] for P slices in the H.264/AVC video coding standard. Our scheme initially exploits neighbourhood information jointly with a set of skip mode conditions for enhanced skip mode decision making. It subsequently performs inter mode decision for the remaining macroblocks by using a gentle set of smoothness constraints. For RD performance very close to the standard we achieve 35-58% reduction in run times and 33-55% reduction in CPU cycles for both the rate controlled and the non rate controlled versions of H.264/AVC. Compared to other work that has been proposed as input to the H.264/AVC standard activity, gains of 9-23% in run times and 7-22% in CPU cycles are also reported.

4.2 Observations on Previous Work

A careful evaluation of the percentage of macroblock modes in the H.264/avc standard across sequences with different motion characteristics in P slices, reveals that the SKIP mode dominates among other modes (particularly at low bit rates). This is shown in the table below from the rate controlled version of the JM 8.2 model:

Sequences	*45k rate*	*30k rate*	*20k rate*	*10k rate*
News	66%	71%	77%	92%
Silent	59%	65%	74%	91%
Container	58%	66%	72%	86%
Salesman	56%	74%	79%	95%
Bridge-far	62%	75%	83%	90%
Grandma	63%	68%	73%	86%
Foreman	24%	35%	51%	76%
Car-phone	33%	43%	56%	86%

Table 4-1 Percentage of Macroblocks in Skip Mode for Different Video Sequences

The above observation implies that accurate detection of the SKIP mode type without performing the computationally intensive mode decision would be highly desirable for computational speed-ups. Obviously we can say that fast motion video sequences like Stefan or Mobile will have even lower percentage of skip modes compared with slow motion video sequences and the computational gains for predictive SKIP mode schemes are indeed content dependent, since slow motion sequences or sequences with uniform motion fields tend to have more skipped macroblocks.

Previous work on SKIP mode prediction for P Slices has been mentioned in section 3.2.2 (Skip Mode Detection, page 41-42). According to this work [55], a block can be predicted as having SKIP mode in a P slice when the following set of four conditions is satisfied. This set of four conditions is not sufficient due to the assumption that the mode with the lowest RD cost is the inter MODE_16*16 (condition 1), which may or may not be true. If it is true, then we can safely say that the macroblock can be skipped since $J_{SKIP} < J_{MODE\_16*16}$ for the same motion vectors as condition 3 above, thus making the SKIP mode the chosen one due to its lowest cost. If the first condition above is not true though, the algorithm will mis-predict macroblocks as skipped and the RD performance will suffer. The important point here is that although condition 1 makes the set of the above conditions *non sufficient* for SKIP mode prediction, it is "good enough" (dependent on the video content of course) as the following table shows:

Sequences	*Percentage*
News	71.49%
Silent	70.12%
Container	65.86%
Grandma	70.06%
Mother_and_Daughter	61.37%
Salesman	73.24%
Bridge-far	82.44%
Bridge-close	83.03%
Carphone	61.46%
Foreman	57.92%

Table 4-2 Percentage of Times the INTER_16*16 Mode has the lowest RD cost (At 45k bit rates)

Similar percentages were obtained in a range of different bit rates. The above table suggests that computational complexity can indeed be reduced by only examining the RD cost function of the inter MODE_16*16 for skip mode decision. Theoretically, this is further supported from the fact that the condition 4 above is a relatively strong one, since the likelihood of having non zero quantised coefficients in smaller block sizes is very small, if these are all zero in a 16x16 block size. Since the four conditions for SKIP mode decision in section 3.2.2 (page 41-42) is "good enough", we will use it as part of the layered scheme we propose.

4.3 Proposed Scheme

We actually intend to reduce the computation cost even further by predicting a percentage of macroblocks in SKIP mode (under appropriate conditions) based on spatio-temporal neighbourhood information and without any edge detection. In this manner, we can partially avoid the computational cost required for the RD function evaluation required in the first condition above. The idea of using neighbourhood spatio-temporal information is based on the observation (which is particularly evident in slow moving sequences) that the areas of a picture that consist of macroblocks in SKIP mode, slowly change over time as shown in the figure 4-1 below (using two real video pictures).

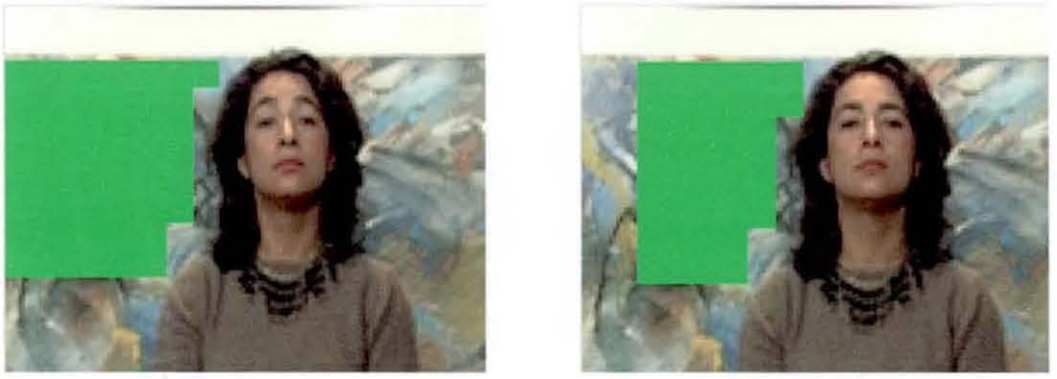


Figure 4-1 Slowly Changing Picture Areas Consisting of Macroblocks in Skip Mode

This suggests that if the macroblocks on the top and left of the one to be encoded in the current frame and the macroblocks on the right and bottom of the co-located macroblock in the previous frame (the right and bottom macroblocks in current frame are not available now, therefore we choose the ones at same locations in previous frame to be good substitutes.) are all in skip mode, then this mode pattern can be a good indication that the current macroblock can be skipped. This is due to the fact that the likelihood of the current macroblock belonging to a stationary (or uniformly moving) part of the picture, and thus being a candidate for skipping, is high based on the above pattern. The elements of the proposed spatio-temporal predictor are shown in the figure below. Co is the collocated macroblock in the reference picture, CoR is the macroblock to the right of the collocated, CoB the one to its bottom, C denotes the macroblock to be mode predicted, L the one to its left and T the one on its top in the current picture:

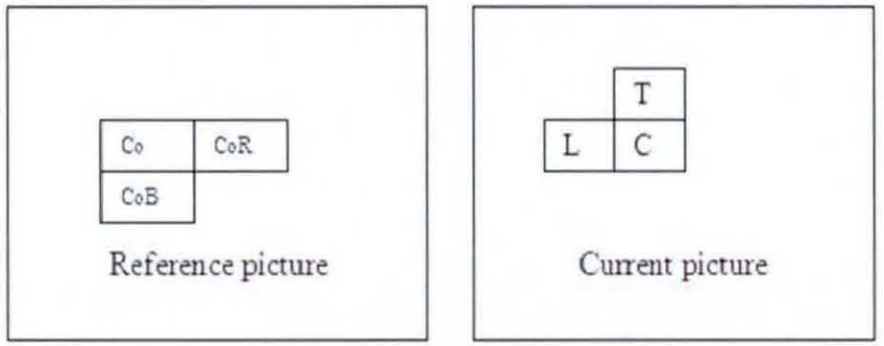


Figure 4-2 Spatio-temporal Predictor for Skip Mode Decision

To strengthen the accuracy of our skip mode prediction, we also add the extra

condition that the Sum of Absolute Differences (SAD) between the current macroblock and its co-located one should be less than the average SAD among the skipped macroblocks in the reference picture and their co-located predictors. Having such an adaptive threshold in fact imposes extra constraints on the local change of the motion field (either in direction or in magnitude), since it requires that it is less than the average change of the motion field of skipped areas (areas of the same type) in the reference picture. This average change is expected to vary slowly between pictures as shown in Figure 5.1, particularly for slow or uniform motion sequences. Another advantage of this extra condition is reducing the affect of scene changes which will make wrong mode decision. During the scene change happens, SAD between current MB and its co-located one will become very big, and this condition cannot be satisfied any more. Current macroblock will not be predicted to be SKIP mode. After a percentage of macroblocks is characterised as skipped based on the above predictor, we use the conditions of Jeon et. al [55] to identify even more skipped macroblocks in a low computational cost framework.

For the remaining macroblocks (which also include a percentage of skipped macroblocks that were not classified with the first two stages), we are left with the task of predicting their respective modes while trying to avoid as much computation (for the RD cost evaluation of different modes) as possible. This is not a straight-forward task, since the accuracy versus complexity trade-off is apparent in this prediction process. To be more specific, smoothness techniques based on Markov Random Fields (MRFs) [92], higher order statistics techniques including variance, skewness and kurtosis estimates [93] or directional fields [94] can do a very accurate prediction of macroblock modes in terms of texture properties, but their computational requirements are prohibitive for one pass coding systems. Even the simplest of techniques based on spatial pixel gradients [67, 70] needs to calculate all gradients in the macroblock initially, to determine the angles on the Cartesian plane for each of these gradients, to group these angles into Cartesian plane sections associated with particular macroblock modes and finally to determine as the most probable mode the one corresponding to the plane section with the largest group of angles. The important points in the work of [67, 70] is the ability to predict accurately the coding mode of smooth areas in the macroblock level with spatial information only and that the likelihood of these areas to have the inter MODE_16*16 as their best coding

mode is high. We will use the second of these points in the design of our criteria for mode prediction.

The problem of inter mode prediction of *the remaining macroblocks* in a low computational cost framework can be decomposed in two sub-problems, namely in the problems of decidable modes under smoothness conditions and on un-decidable ones under the same conditions. Decidable modes include SKIP, MODE_16*16, MODE_16*8, MODE 8*16, INTRA16*16 and INTRA4*4 in our scheme, while the un-decidable ones include exhaustive evaluation of RD costs for all candidate modes.

The following three smoothness conditions for decidable modes are proposed in our scheme:

a) The J_{MODE_16*16} cost of the current macroblock should be less than the average J_{MODE_16*16} cost of the macroblocks in this mode *in the reference frame.*

b) The co-located macroblock in the reference frame should be of skip mode *or* of MODE_16*16

c) The SAD between the current and the co-located macroblock should be less than the average SAD among the skip mode macroblocks in the previous frame and their collocated predictors.

For the first condition, we need to stress two points. Firstly, that the RD cost of the MODE_16*16 is known from the second stage of our scheme (skip mode decision), so no extra computation is needed. Secondly, temporal rather than spatial information was chosen for designing this adaptive threshold, since although MODE_16*16 is highly likely for smooth areas [67, 70], it is the temporal rather than the spatial information that is more relevant to the J costs and this information is available without any extra pre-processing apart from a mean calculation (in contrast to the gradient scheme of [67, 70]). For the second condition, it was also deemed that the classification between SKIP and inter MODE_16*16 for the co-located macroblock would require extra processing, so

these modes were not distinguished explicitly. The reasons for the choice of the third smoothness condition are explained in the previous page. This concept of grouping modes together was also proposed in [61, 77, 95] but with different mode groups and with different smoothness conditions (if any), while work in [67, 70] explicitly decides modes by assigning a single mode to a macroblock *with extra pre-processing computation.* Some observations relating to the performance of *mode grouping based schemes* (as the most relevant to ours for comparison purposes) will be mentioned in the experiments section. Our scheme concludes by testing the validity of the three smoothness conditions. If they are satisfied, only 154 out of 171 RD costs are evaluated, else all RD costs are evaluated as the standard (un-decidability under these smoothness conditions). The proposed multi-layer mode decision scheme for P slices can be summarised with the following pseudo-code (Figure 4-3) and the block diagram (Figure 4-4):

If ($MODE_{TOP\_MACROBLOCK\_CURRENT\_FRAME}$, $MODE_{LEFT\_MACROBLOCK\_CURRENT\_FRAME}$,

$MODE_{RIGHT\_MACROBLOCK\_COLLOCATED\_REF\_FRAME}$, $MODE_{LEFT\_MACROBLOCK\_COLLOCATED\_REF\_FRAME}$)
= = SKIP (all of them)
AND SAD(CURRENT_MACROBLOCK,COLLOCATED) <
AVERAGE SAD (SKIPPED_MACROBLOCKS_REF_FRAME, COLLOCATED PREDICTORS)

$MODE_{CURRENT\_MACROBLOCK} = SKIP$

If for the remaining macroblocks:
((The lowest RD cost for the current macroblock is for MODE_16*16) AND
(There is a single reference frame) AND (The best motion vector is the
Predicted motion vector (including the zero MV)) AND
(The transform coefficients of the current macroblock are all quantised to zero))

$MODE_{CURRENT\_MACROBLOCK} = SKIP$

If for the remaining macroblocks:
((The J_{MODE_16*16} cost of the current macroblock should be less than the
average J_{MODE_16*16} cost of the macroblocks in this mode *in the reference frame*)
AND (The co-located macroblock in the reference frame should be of SKIP
mode *or* MODE_16*16)
AND SAD(CURRENT_MACROBLOCK,COLLOCATED) <
AVERAGE SAD (SKIPPED_MACROBLOCKS_REF_FRAME, COLLOCATED PREDICTORS)

$MODE_{CURRENT\_MACROBLOCK} \in \{SKIP$, MODE_16*16, MODE_16*8, MODE 8*16,
INTRA16*16,INTRA4*4$\}$

For all the remaining macroblocks:

Figure 4-3 Pseudo Code of the Algorithm

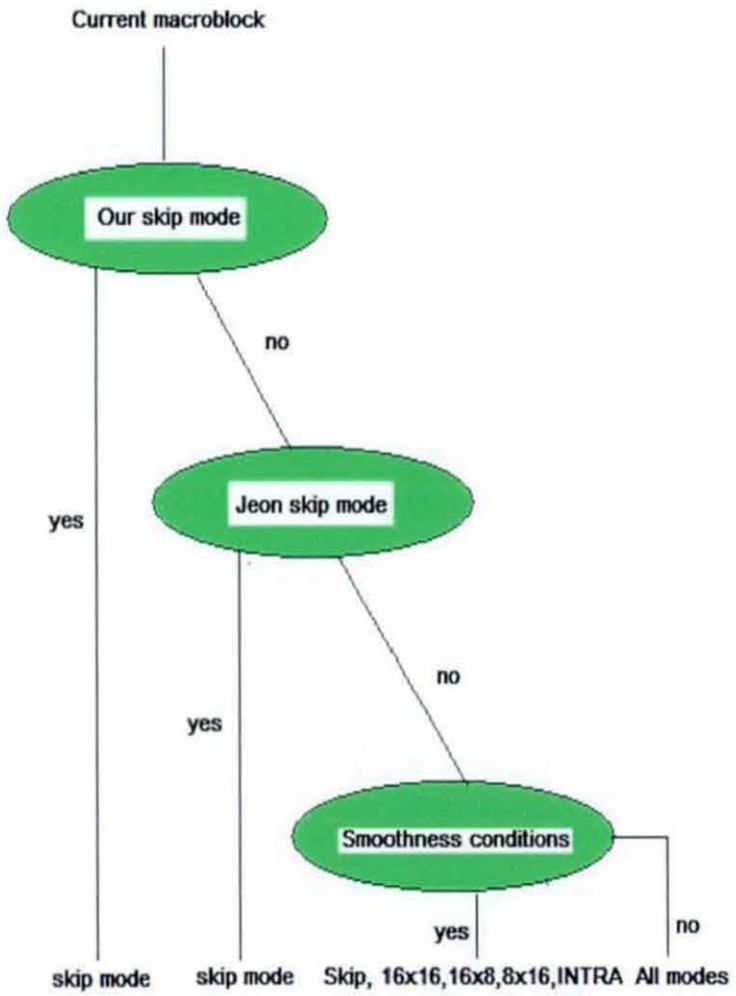


Figure 4-4 Flow Chart of the Algorithm

4.4 Experiments and Discussion

To assess the proposed scheme, a comprehensive set of experiments for a variety of video sequences with different motion characteristics was performed on a Pentium 4

(3.06 GHz processor). In particular, the exact characteristics of the test sequences are summarised in the following paragraph.

"News" is a sequence of low spatial detail that contains medium changes in motion. "Silence" is a sequence of low spatial detail and medium changes in the motion of the arms and head of the person in the sequence. "Container" is a sequence with dominant horizontal motion (fast in segments of the video) but also it contains a mixture of oscillating and slow motion in part of the frames. "Salesman" and "Grandma" are typical video-conferencing sequences with low spatial detail and slow motion. "Bridge-far" is a far view of a bridge with people crossing. It contains very slow horizontal and oscillating motion. Sequences of similar nature are encountered in surveillance videos of public places. "Foreman" is a sequence with medium changes in motion and contains dominant luminance changes. Finally "car phone" contains medium foreground motion with fast background panning motion.

Both the cases of rate controlled and non rate controlled mode prediction were examined in the JM 8.2 model of the standard. In the set of metrics, we included both run times (in seconds) and cycle counts from a static analysis of the Pentium 4 processor. It has to be noted that the static analysis does not measure run time program behaviour that benefits from cache locality and thus the gains are expected to be lower, albeit useful for embedded processor designers. Furthermore, according to our observations run times are more meaningful from the user's point of view. We compared our results with the standard and with the algorithm of Jeon et. al [55], which was proposed as input to the standard and is based on the set of skip mode conditions of Section 3.2.2. Comparisons were made in terms of bitrates reduction (Δbitrates), PSNR gain (ΔPSNR), run time savings (%) and cycle savings (%) for a set of fixed quantisation parameters {28, 32, 36, 40} for the non rate controlled case and for target bit rates 45, 30, 20 and 10 kilobits for the rate controlled case in table 4-3 to 4-10. We also give graphs (from 4-1 to 4-16) which are the R-D curves among original, JEON's and our algorithms. From these graphs, we can see that there are only very little difference between the three curves, and this shows that our algorithm does not degrade the coding efficiency and picture quality very much. The configuration file for the encoder had the following settings: Main Profile,

Level 30, RD optimisation ON, Hadamard Transform, IPPP... structure, CABAC coding, no error tools, number of reference frames was 5 and Search Range was 16. For rate control, the initial QP was 24 and the number of basic units 11.

QP=28				
JEON with respect to H.264/AVC				
Sequences	ΔPSNR	Δ Bitrates	Time Savings (%)	Cycle Savings (%)
news	-0.05	-0.07	39.22	37.61
silent	-0.01	-0.08	33.23	32.23
Container	-0.03	-0.65	38.43	36.18
Salesman	0	-0.57	48.58	46.36
Bridge-far	-0.01	0	62.71	59.75
Grandma	-0.02	-0.90	40.78	39.34
Foreman	0.01	0.09	8	6.48
Car phone	-0.05	-0.25	9.19	8.72
Our scheme with respect to H.264/AVC				
Sequences	Δ PSNR	Δ Bitrates	Time Savings (%)	Cycle Savings (%)
news	-0.05	-0.29	43.40	41.45
silent	-0.03	-0.62	37.68	36.39
Container	-0.04	-0.11	45.04	42.75
Salesman	0	-0.22	54.88	52.58
Bridge-far	-0.02	0	72.82	69.81
Grandma	-0.02	-0.78	49.01	47.31
Foreman	0	0.67	12.5	11.47
Car phone	-0.05	-0.12	13.21	13.11

Table 4-3 ΔPSNR/ΔBitrates/Time saving/Cycles saving of JEON and Our Schemes with respect to H.264/AVC (QP=28)

QP=32				
JEON with respect to H.264/AVC				
Sequences	Δ PSNR	Δ Bitrates	Time Savings (%)	Cycle Savings (%)
news	-0.05	-0.21	42.95	41.10
silent	-0.01	0.28	36.28	34.50
Container	-1.04	-1.17	42.66	40.69
Salesman	-0.05	-0.66	44.62	43.36
Bridge-far	0	0	69.44	66.53
Grandma	-0.01	-0.57	42.95	41.36
Foreman	-0.01	-0.63	9.48	9.029
Car phone	-0.01	0.13	12.38	11.58
Our scheme with respect to H.264/AVC				
Sequences	Δ PSNR	Δ Bitrates	Time Savings (%)	Cycle Savings (%)
news	-0.03	-0.64	48.32	45.79
silent	-0.01	0.03	42.98	41.00
Container	-0.05	-1.10	50.33	48.65
Salesman	-0.03	-1.04	58.54	56.58
Bridge-far	0	0	78.96	75.90
Grandma	-0.03	-0.49	54.29	52.16
Foreman	-0.01	-0.72	15.44	15.35
Car phone	-0.01	-0.33	17.64	17.16

Table 4-4 ΔPSNR/ΔBitrates/Time saving/Cycles saving of JEON and Our Schemes with respect to H.264/AVC (QP=32)

QP=36				
JEON with respect to H.264/AVC				
Sequences	ΔPSNR	Δ Bitrates	Time Savings (%)	Cycle Savings (%)
News	0	-0.30	40.50	39.14
Silent	0	-0.37	33.33	32.23
Container	-0.03	-1.52	43.57	42.09
Salesman	0	-0.17	43.72	42.44
Bridge-far	0	0	63.11	60.11
Grandma	0.04	-1.49	49.63	47.83
Foreman	0	-0.11	12.71	12.24
Car phone	-0.04	0.22	16.38	15.55
Our scheme with respect to H.264/AVC				
Sequences	ΔPSNR	Δ Bitrates	Time Savings (%)	Cycle Savings (%)
News	-0.03	-0.56	48.02	46.13
Silent	-0.01	-0.37	44.98	43.66
Container	-0.02	-1.52	56.42	54.25
Salesman	0.01	-0.25	60.77	58.96
Bridge-far	0	0	76	72.49
Grandma	0	-2.99	60.94	59.16
Foreman	0	-0.07	19.36	19.18
Car phone	0.02	0.311	22.07	21.34

Table 4-5 ΔPSNR/ΔBitrates/Time saving/Cycles saving of JEON and Our Schemes with respect to H.264/AVC (QP=36)

QP=40				
JEON with respect to H.264/AVC				
Sequences	ΔPSNR	Δ Bitrates	Time Savings (%)	Cycle Savings (%)
News	-0.07	-0.42	37.87	36.74
Silent	-0.02	-0.09	35.93	34.66
Container	0.03	0.77	40.46	39.34
Salesman	0.05	0.47	47.69	46.18
Bridge-far	0	0	60.39	57.24
Grandma	0.01	-0.52	48.60	46.46
Foreman	0.01	0.17	14.90	14.31
Car phone	0	0.07	21.70	20.57
Our scheme with respect to H.264/AVC				
Sequences	ΔPSNR	Δ Bitrates	Time Savings (%)	Cycle Savings (%)
News	-0.03	0.33	51.13	49.00
Silent	0	-0.66	50.50	48.47
Container	-0.01	0.19	56.80	54.58
Salesman	0.04	-0.31	65.13	62.83
Bridge-far	0	0	73.76	70.08
Grandma	0.01	0.26	65.33	62.80
Foreman	-0.03	0.11	22.67	22.38
Car phone	0.04	0.60	28.82	27.77

Table 4-6 ΔPSNR/ΔBitrates/Time saving/Cycles saving of JEON and Our Schemes with respect to H.264/AVC (QP=40)

Target=45k bits				
JEON with respect to H.264/AVC				
Sequences	**ΔPSNR**	**Δ Bitrates**	**Time Savings (%)**	**Cycle Savings (%)**
News	0.03	0.13	40.14	38.41
Silent	0.01	-0.11	35.54	33.68
Container	0.03	0.22	29.51	28.68
Salesman	-0.05	-0.67	44.52	43.03
Bridge-far	0.01	0	25.29	24.49
Grandma	0.04	-0.11	34.88	33.76
Foreman	0.04	0.35	8.26	8.24
Car phone	-0.6	-0.08	10.28	9.98
Our scheme with respect to H.264/AVC				
Sequences	**ΔPSNR**	**Δ Bitrates**	**Time Savings (%)**	**Cycle Savings (%)**
News	-0.01	0.06	43.43	41.21
Silent	0.01	-0.02	38.87	36.58
Container	0.02	0	36.38	34.72
Salesman	-0.04	-0.80	50	47.88
Bridge-far	0	0.08	39.58	38.11
Grandma	0.01	0.022	41.19	39.66
Foreman	0.01	0.33	14.81	14.13
Car phone	-0.05	0	15.88	15.40

Table 4-7 ΔPSNR/ΔBitrates/Time saving/Cycles saving of JEON and Our Schemes with respect to H.264/AVC (45kbits)

Target=30k bits				
JEON with respect to H.264/AVC				
Sequences	**ΔPSNR**	**Δ Bitrates**	**Time Savings (%)**	**Cycle Savings (%)**
News	-0.03	-0.36	40.49	39.38
Silent	-0.01	0.72	36.29	35.42
Container	0.05	-0.06	37.02	35.60
Salesman	0	-0.58	49.04	47.19
Bridge-far	-0.12	0.23	33.63	32.59
Grandma	0.03	0.13	37.01	35.59
Foreman	-0.02	0.13	11.56	11.31
Car phone	0.03	-1.11	14.58	14.26
Our scheme with respect to H.264/AVC				
Sequences	**ΔPSNR**	**Δ Bitrates**	**Time Savings (%)**	**Cycle Savings (%)**
News	-0.06	0	44.62	42.85
Silent	-0.04	0.46	41.48	39.55
Container	0.02	-0.19	43.98	42.05
Salesman	0.02	-0.58	53.61	51.35
Bridge-far	-0.14	0.10	47.70	46.18
Grandma	0.01	0.03	44.83	43.48
Foreman	0.05	0.19	18.75	18.20
Car phone	-0.01	-1.34	20.13	19.25

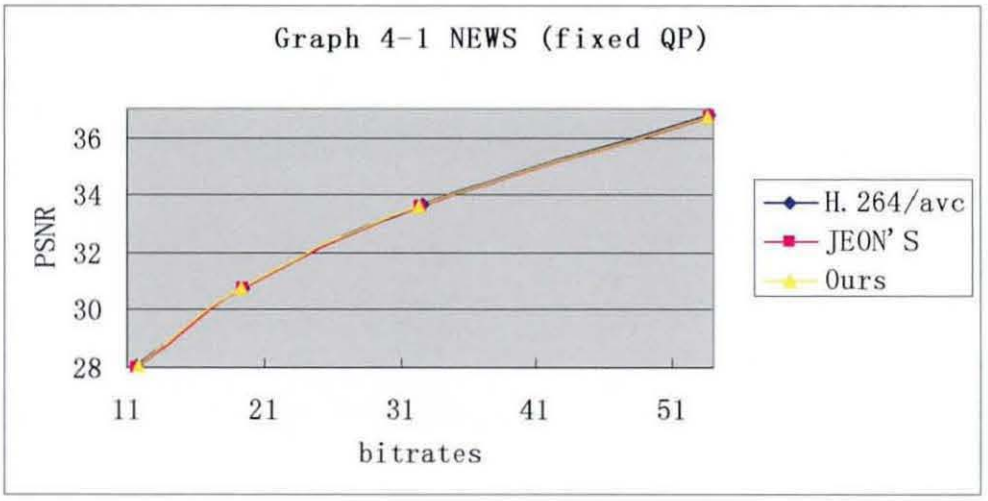
Table 4-8 ΔPSNR/ΔBitrates/Time saving/Cycles saving of JEON and Our Schemes with respect to H.264/AVC (30kbits)

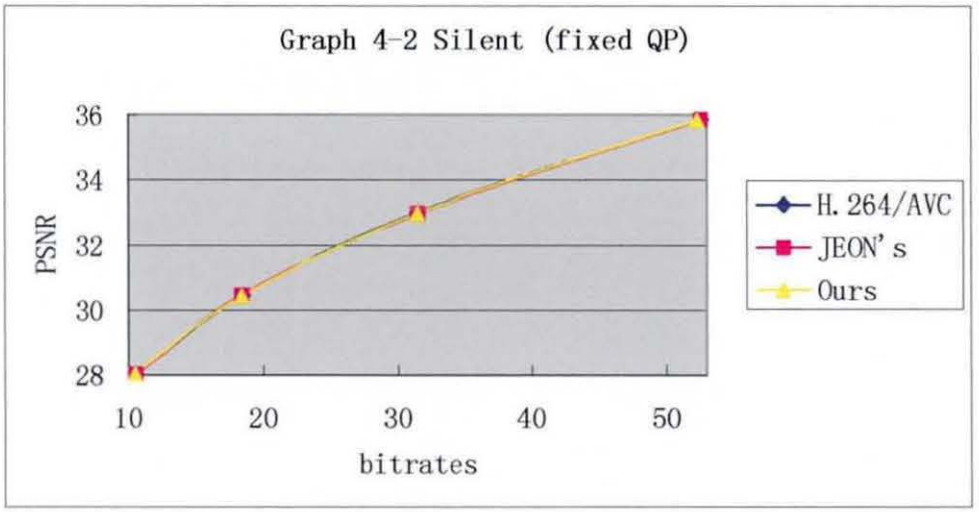
Target=20k bits				
JEON with respect to H.264/AVC				
Sequences	ΔPSNR	Δ Bitrates	Time Savings (%)	Cycle Savings (%)
News	0.09	0	43.98	42.43
Silent	-0.03	0.09	40.66	38.95
Container	0	0.24	42.19	40.76
Salesman	0.01	0.44	48.29	46.44
Bridge-far	-0.17	-0.14	39.48	38.34
Grandma	-0.05	0.29	39.76	38.26
Foreman	-0.02	-0.04	15.14	14.84
Car phone	0.02	0.77	17.05	16.39
Our scheme with respect to H.264/AVC				
Sequences	ΔPSNR	Δ Bitrates	Time Savings (%)	Cycle Savings (%)
News	0.08	0.049	47.68	45.38
Silent	-0.02	-0.04	46.05	43.68
Container	-0.02	0.34	50.35	48.08
Salesman	0.1	0.34	55.55	52.96
Bridge-far	-0.21	-0.09	53.72	51.76
Grandma	0.01	0.34	50.57	48.52
Foreman	0.01	-0.09	23.59	22.69
Car phone	-0.02	0.33	24.41	23.57

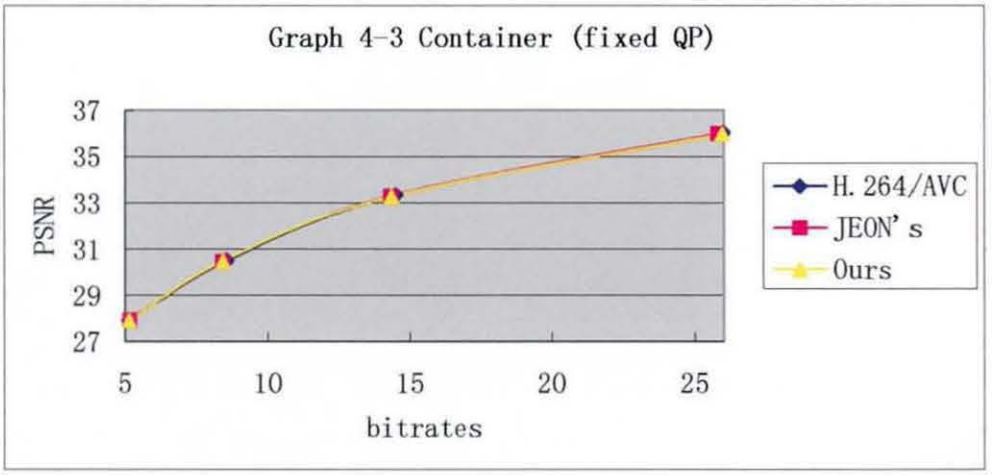
Table 4-9 ΔPSNR/ΔBitrates/Time saving/Cycles saving of JEON and Our Schemes with respect to H.264/AVC (20kbits)

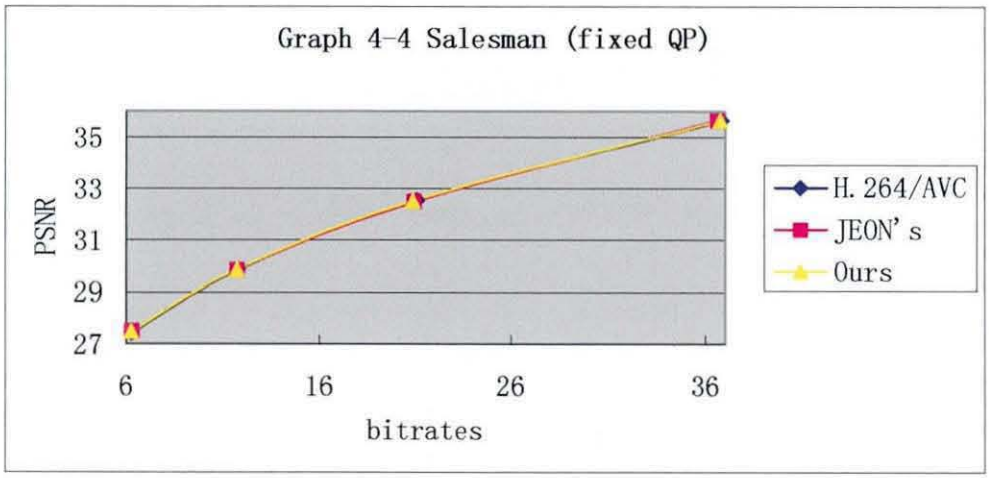
Target=10k bits				
JEON with respect to H.264/AVC				
Sequences	ΔPSNR	Δ Bitrates	Time Savings (%)	Cycle Savings (%)
News	-0.04	-1.13	49.14	45.66
Silent	-0.04	-0.88	55.49	52.54
Container	0.05	0	50	47.11
Salesman	0.01	0	63.80	60.14
Bridge-far	-0.02	-0.69	48.93	47.05
Grandma	0.12	0	48.45	46.13
Foreman	-0.09	-0.59	21.77	21.15
Car phone	-0.03	-0.35	34.18	32.46
Our scheme with respect to H.264/AVC				
Sequences	ΔPSNR	Δ Bitrates	Time Savings (%)	Cycle Savings (%)
News	0.04	-1.04	62.85	58.19
Silent	-0.01	-0.19	62.63	58.78
Container	0.02	0	60.78	57.57
Salesman	-0.1	0	72.39	67.49
Bridge-far	0.04	-0.89	61.70	59.73
Grandma	0.02	0	57.73	54.60
Foreman	-0.15	0	34.22	32.74
Car phone	-0.01	-0.61	49.48	46.92

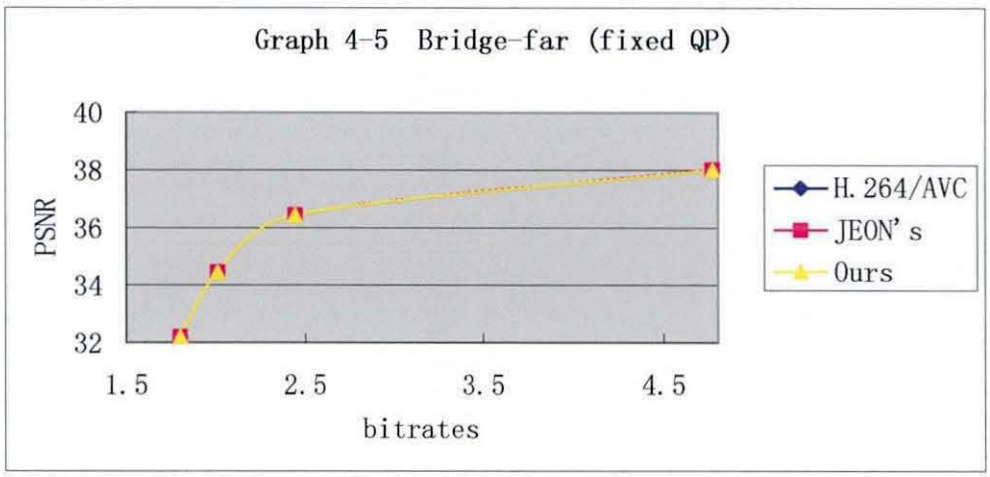
Table 4-10 ΔPSNR/ΔBitrates/Time saving/Cycles saving of JEON and Our Schemes with respect to H.264/AVC (10kbits)

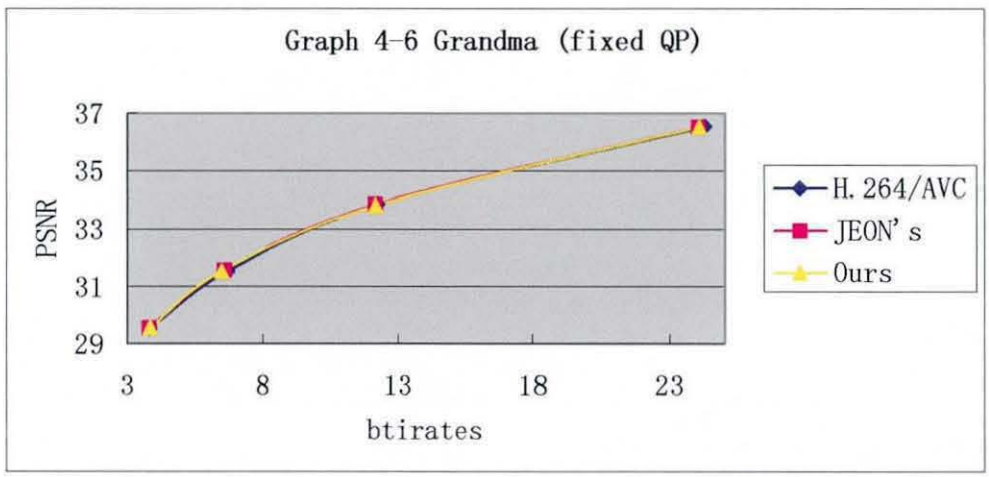

Graph 4-1 NEWS (fixed QP)
PSNR
36
34
32
30
28
11
21
31
41
51
bitrates
H. 264/avc
JEON' S
Ours

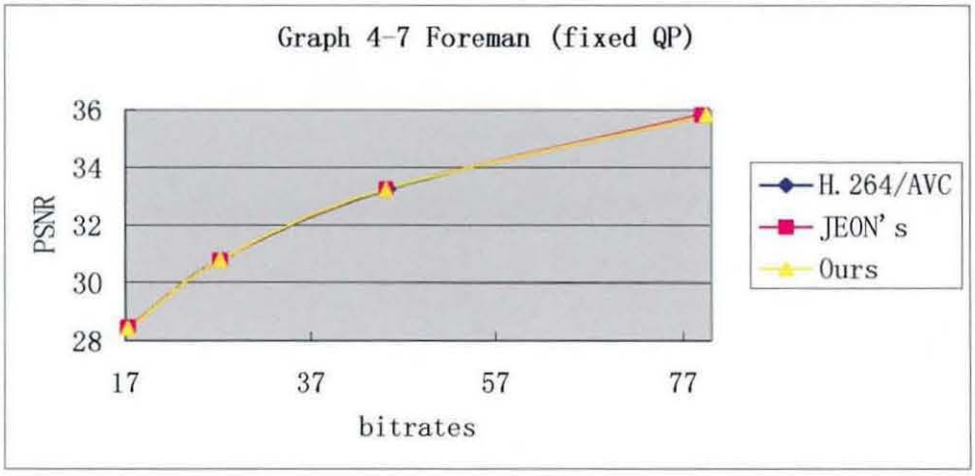

Graph 4-2 Silent (fixed QP)
PSNR
36
34
32
30
28
10
20
30
40
50
bitrates
H. 264/AVC
JEON' s
Ours

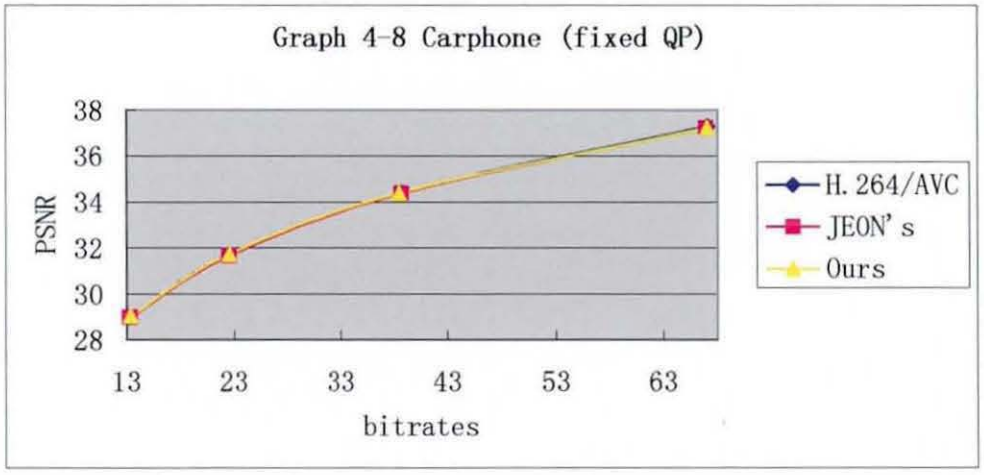

Graph 4-3 Container (fixed QP)
PSNR
37
35
33
31
29
27
5
10
15
20
25
bitrates
H. 264/AVC
JEON' s
Ours

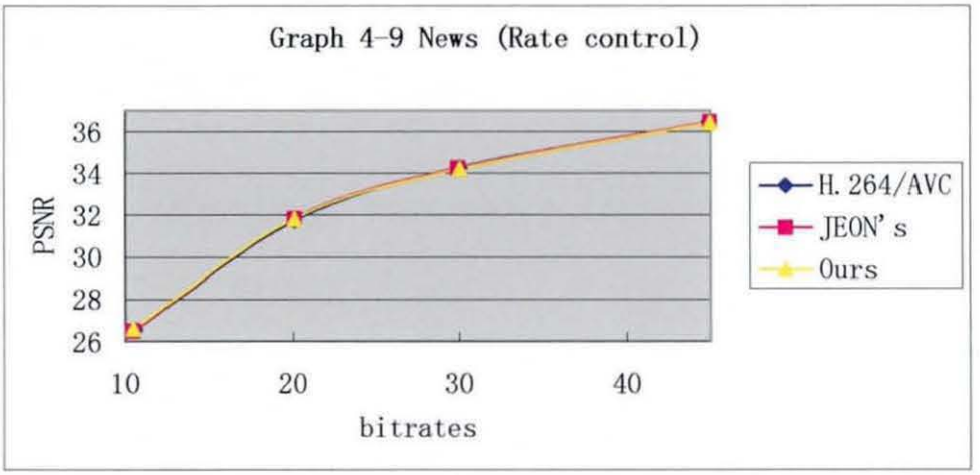

Graph 4-4 Salesman (fixed QP)
PSNR
35
33
31
29
27
6
16
26
36
bitrates
H. 264/AVC
JEON' s
Ours

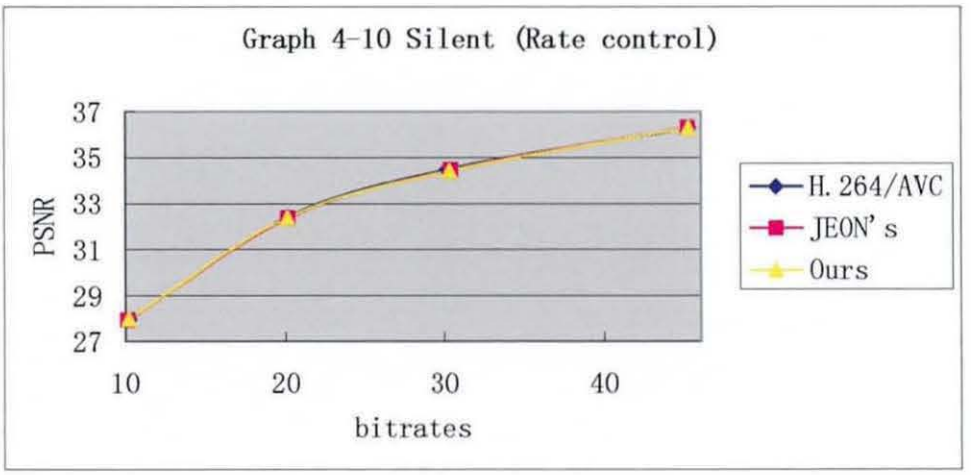

Graph 4-5 Bridge-far (fixed QP)
PSNR
40
38
36
34
32
1.5
2.5
3.5
4.5
bitrates
H. 264/AVC
JEON' s
Ours

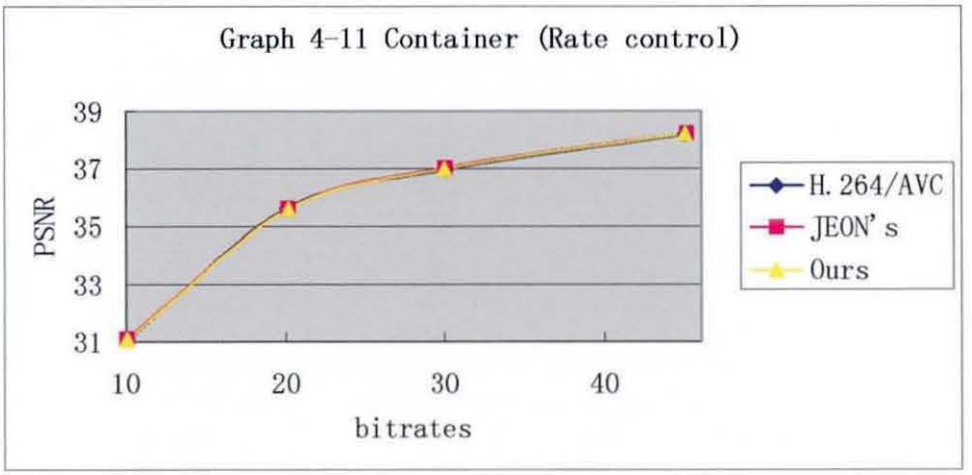

Graph 4-6 Grandma (fixed QP)
PSNR
37
35
33
31
29
3
8
13
18
23
btirates
H. 264/AVC
JEON' s
Ours

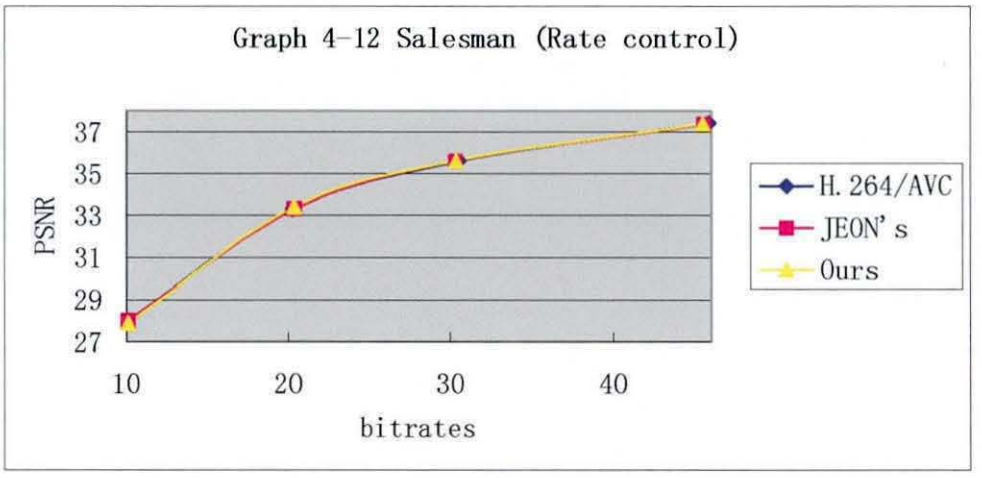

Graph 4-7 Foreman (fixed QP)
PSNR
36
34
32
30
28
17
37
57
77
bitrates
H. 264/AVC
JEON' s
Ours

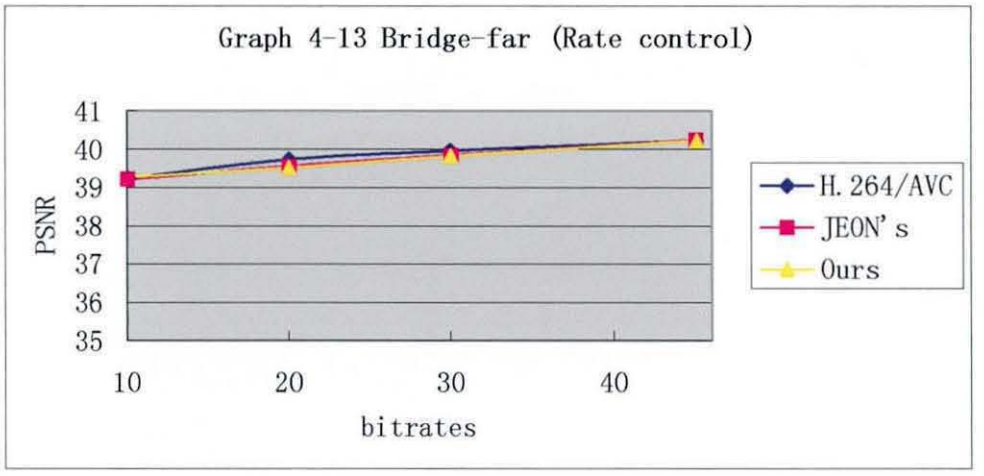

Graph 4-8 Carphone (fixed QP)
PSNR
38
36
34
32
30
28
13
23
33
43
53
63
bitrates
H. 264/AVC
JEON' s
Ours

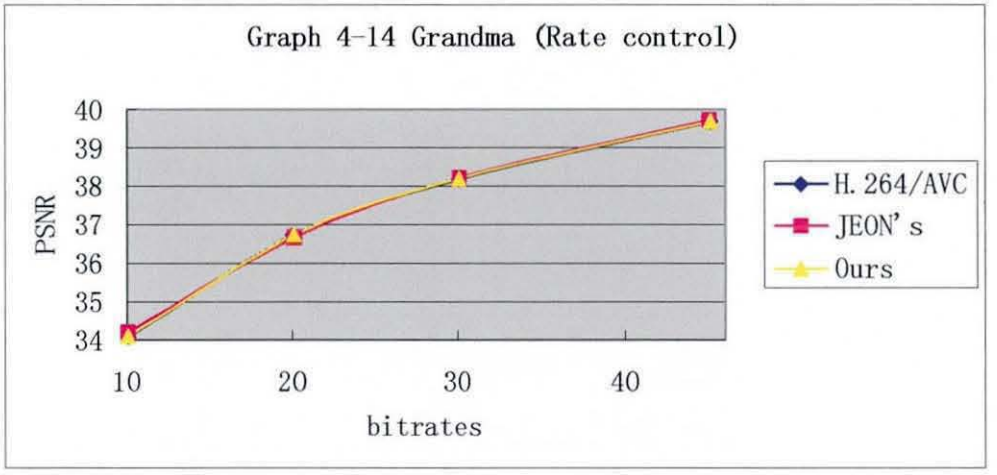

Graph 4-9 News (Rate control)
PSNR
36
34
32
30
28
26
10
20
30
40
bitrates
H. 264/AVC
JEON' s
Ours

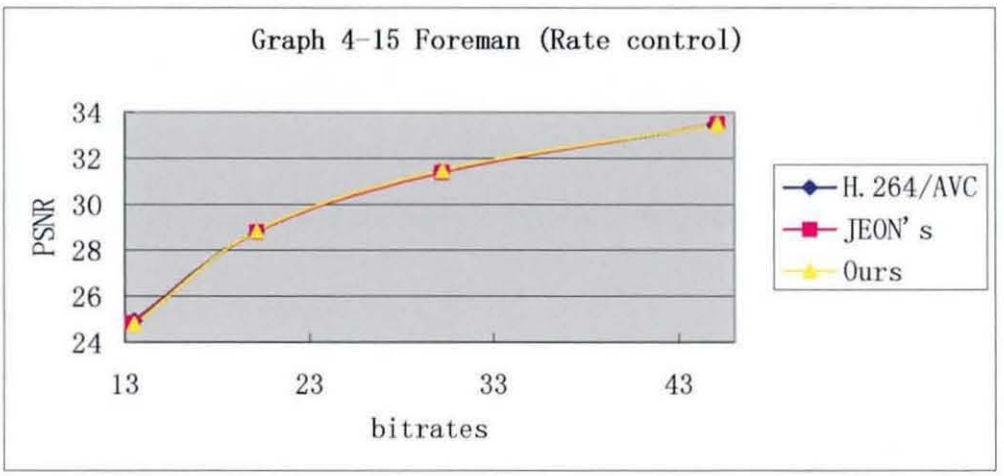

Graph 4-10 Silent (Rate control)
PSNR
37
35
33
31
29
27
10
20
30
40
bitrates
H. 264/AVC
JEON' s
Ours

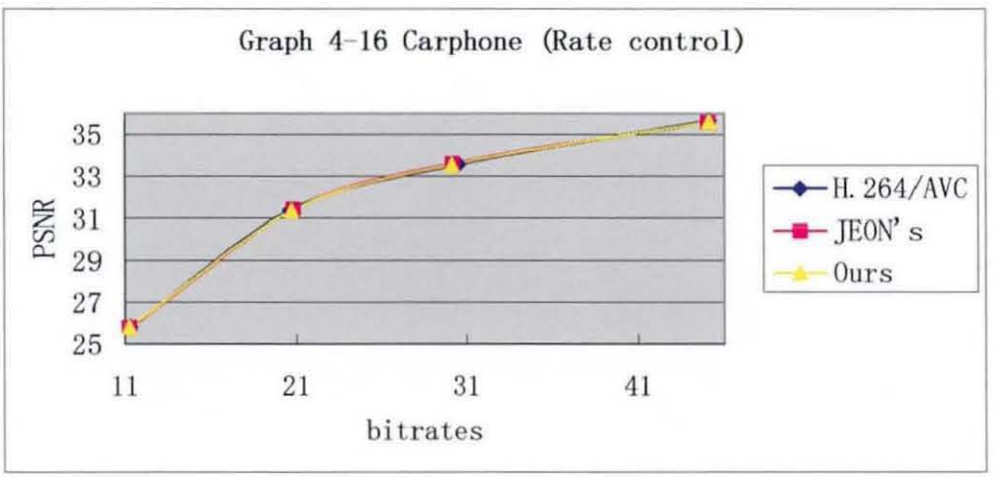

Graph 4-11 Container (Rate control)
PSNR
39
37
35
33
31
10
20
30
40
bitrates
H. 264/AVC
JEON' s
Ours


Graph 4-12 Salesman (Rate control)
PSNR
37
35
33
31
29
27
10
20
30
40
bitrates
H. 264/AVC
JEON' s
Ours


Graph 4-13 Bridge-far (Rate control)
PSNR
41
40
39
38
37
36
35
10
20
30
40
bitrates
H. 264/AVC
JEON' s
Ours


Graph 4-14 Grandma (Rate control)
PSNR
40
39
38
37
36
35
34
10
20
30
40
bitrates
H. 264/AVC
JEON' s
Ours


Graph 4-15 Foreman (Rate control)
PSNR
34
32
30
28
26
24
13
23
33
43
bitrates
H. 264/AVC
JEON' s
Ours


Graph 4-16 Carphone (Rate control)
PSNR
35
33
31
29
27
25
11
21
31
41
bitrates
H. 264/AVC
JEON' s
Ours

From the results, we could see that time savings and cycles savings depend not only on video contents but also on quantisation parameters and target bitrates. Our algorithms can achieve more computational reduction in slow motion video sequences with respect to fast motions ones. For example at QP 28, our algorithm can attain 43% time savings in News but only 11.5% in Foreman. At higher quantisation parameters and lower target bitrates more macroblocks are forced to be SKIP mode or bigger block size mode which are satisfied by SKIP mode conditions or smooth constrains, therefore time and cycles savings can be bigger comparing with lower quantization or higher target bitrates. In Silent we can achieve 50.5% time savings at QP40 but only 37% at QP 28, 62.6% at 10k but only 38.9% at 45k. For fixed quantisation parameters the average gains in run times for QP=28 are 40%, for QP=32 are 46% and for QP=40 are 52% with respect to the standard. As compared to the algorithm in [55], the average gains are 11%, 14.3% and 22% for the same order of quantisation parameters. In terms of CPU cycles (again for the same order of QPs) the average gains are 39.3%, 44% and 50% respectively with respect to the standard and 10.3%, 14.3% and 22% with respect to the work in [55]. For target rates of 45k bits, the average gains in run times and CPU cycles are 35% and 33.5% with respect to H.264/AVC, while 9% and 7% are the gains with respect to [55]. For target rates of 20k bits, the average gains in run times and CPU cycles are 44% and 42% with respect to H.264/AVC, while 13% and 11.3% are the gains with respect to [55]. For target rates of 10k bits, the average gains in run times and CPU cycles are 58% and 54.5% with respect to H.264/AVC, while 21.3% and 19% are the gains with respect to [55]. It can also be seen from graph 5-1 to 5-16 that the RD performance of our scheme is very similar to the standard and to the work in [55] for all cases examined.

A careful analysis of the results reveals that our algorithm performs better in terms of run time/CPU cycle gains for lower target bit rates in the rate controlled cases and for higher quantisation step sizes in the non rate controlled ones. This is due to the increase in the number of skipped macroblocks. It also favours sequences with slow or uniform motion but it does not penalise ones with fast or abnormal motion changes since the conditions for mode decision based on smoothness constraints are gentle. In fact, due to the enhanced skip conditions using local neighbourhood information, our scheme not

only does not suffer in RD performance due to macroblock skipping but it also accurately identifies potential areas of skipped macroblocks in sequences with mixed motion or luminance changes. The worst case performance of our algorithm (i.e. inability to skip macroblocks based on the enhanced conditions and inability to reduce RD evaluations when the smoothness conditions are not satisfied) is the performance of the standard, as can be seen from the pseudo-code in figure 4-3.

It is worth mentioning that although in the proposed scheme we do not deal explicitly with intra/inter mode decision (we rather estimate the macroblock mode from a mixed intra/inter mode group) , our scheme would be theoretically faster if it included an inter/intra mode decision condition as compared to the second scheme proposed in [55]. In that work, the choice between inter and intra modes is performed after the RD costs of all inter modes are evaluated in order to identify the best one and after the Average Boundary Error (ABE) is computed for the neighbouring macroblocks in the current frame. The algorithm then decides to proceed in the intra mode RD evaluation only if the texture rate of the best inter mode is greater than k*ABE (k is a user defined threshold dependent on the sequence), potentially requiring only 23 out of 171 RD cost evaluations. In our scheme, we would only require 20 RD cost evaluations *on a reduced set of macroblocks* (due to our enhanced skip conditions) since we would still have to account for SKIP mode and we would not have to account for 8*8 modes, according to our smoothness criteria described in Section 4.3.

As compared to the work in [77], our scheme has benefits since it uses neighbourhood information for enhanced skipping, the thresholds for mode decision are soft and content adaptive and generally the smoothness conditions for mode decision are gentler thus promoting a better RD performance. The authors in [77] do a grouping of macroblock modes similar to the one in later versions of the H.263 standard but their objective is to jointly optimize the mode decision and the sub-pixel motion estimation processes, whereas we are only attempting to speed-up the mode decision process. We found that their use of hard thresholds for mode decision has benefits in terms of time savings but it has a negative impact on the RD performance. For example for fast moving sequences like "Stefan" or for sequences with luminance changes such as "foreman" they

report average PSNR losses greater than 0.1 dB for higher time savings, whereas our PSNR losses are smaller than that for a whole range of sequences and sometimes our average PSNR is even better than the one of H.264/AVC. This implies that we are looking for a gentler trade-off between time savings and distortion performance. Furthermore, the use of hard thresholds (constants) is highly dependent on the video content and potentially difficult to tailor. The authors in [95] attempt a probabilistic analysis of a cost estimate and they derive thresholds based on maximum likelihood estimators. Although the gains achieved are significant in terms of RD performance versus time, from its experiments we can say its thresholds are attained by training video sequences before real video encoding, therefore we believe that their approach is impractical in one pass coding systems with unknown video content probability distributions and computational constraints. Finally, a comparison of our work with [61] (which outperforms the approaches in [59, 60]) reveals again better RD performance, since average bit rate increases of 9.8% for the same average PSNR as the standard are reported.

4.5 Conclusion

We have proposed an inter mode decision scheme for P slices that exploits neighbourhood information along with a set of skip mode conditions for enhanced skip mode decision and by using a gentle set of smoothness constraints it also performs inter mode decision for the remaining macroblocks. For RD performance very close to the standard, the proposed scheme affords significant time/cycle savings as compared to H.264/AVC and other known work that was proposed as input to the standard. Our scheme is very relevant to low complexity coding systems.

Chapter 5

Fast Mode Prediction for the Baseline and Main Profiles

5.1 Preamble

We further propose a novel fast mode decision scheme [97] which can be considered as an extension of previous algorithm in Chapter 4 for both the baseline and main profiles of the H.264/AVC video coding standard. Our scheme combines a set of conditions for SKIP mode decision, two heuristics that reduce the set of inter modes to be examined, inter/intra mode prediction and the monotonicity property of the cost functions. For very similar RD performance, we achieve (33%-90%) reduction in run times as compared to H.264/AVC standard. Compared to other work that was used as input to the standard, our scheme is faster by 9-23% for very similar RD performance. The proposed scheme can be used either in its entirety or as individual components for speeding up the standard. It can be also used in conjunction with fast motion search methods and is applicable in both the rate controlled and non rate controlled cases of H.264/AVC standard.

5.2 Observations on Previous work

Table 4-1, 4-2 and our observations of the previous section already show that [55], although not sufficient, is a good skip mode prediction technique for baseline profile. With the same reasoning, we can observe that the set of conditions for SKIP mode

decision in the B slices [56] of the main profile is non sufficient either. This is due to the fact that no motion compensation takes place as can be seen in conditions 1' and 2' (mentioned in 3.2.2, page 41-42) and it is implicitly assumed that the DIRECT_16*16 mode is the one with the minimal cost. If true, the RD performance is not hampered, otherwise it is. Condition 1' in fact enforces the mode of the sub-blocks to be DIRECT_8*8 and the mode of the whole macroblock to be DIRECT_16*16. Furthermore, $J_{SKIP} < J_{MODE\_16*16}$ for the same motion vectors and reference slices from condition 1', clearly implying that the macroblock can be safely skipped. The reader should note that in contrast to the baseline profile, there is actually no assumption in terms of the best reference slice for the SKIP mode detection in the B slices of the main profile. From the design of the standard, the DIRECT and SKIP modes both have the same reference slices and motion vectors, while they are different only in the fact that the SKIP mode needs to have all quantised coefficients in the 8x8 sub-blocks equal to zero, while some coefficients *are not zero* for the DIRECT mode (condition 2'). Theoretically, condition 2' is a relatively strong one in terms of skipping, since the likelihood of having non zero coefficients in a larger block size (16x16) is very small.

Prediction accuracy of skip mode in [55, 56] for baseline and main profiles in H.264/AVC is shown in table 4-11 for different video sequences.

Sequences	P slices ([55])	B slices ([56])
News	99%	98%
Silent	99%	97%
Container	96%	95%
Foreman	96%	87%
Car-phone	98%	86%

Table 5-1 Percentage of macroblocks predicted correctly using the SKIP mode conditions for P and B slices [55, 56] (100 Slices, CIF, Full search used for RD evaluation, 45kbits target rate)

Similar percentages were obtained in a range of different bit rates and with different search methods. The above table suggests that computational complexity can indeed be

reduced for the baseline profile by only examining the RD cost functions of the inter MODE_16*16. Theoretically, this is further supported from the fact that the condition 4 above is a relatively strong one, since the likelihood of having non zero quantised coefficients in smaller block sizes is very small, if these are all zero in a 16x16 block size. Since the above set of conditions for SKIP mode decision in the baseline profile is "good enough", we will use it as part of the layered scheme we propose. The "adequacy" of conditions 1'-2' is also demonstrated by the second column of Table 5-1 and we will also use these conditions in the layered scheme we propose here. It is interesting to note that since the DIRECT mode is in fact a predicted mode (no motion estimation), the accuracy in the prediction of skipped macroblocks in the main profile *is expected to be less* as compared to the baseline profile and this is evident from Table 5-1.

5.3 Proposed Scheme

The big difference between the baseline and main profiles in the H.264/AVC standard, apart from the application domain, is the utilization of Bi-predictive slices. B slices change the encoding order of video sequences. Because of this change, the temporal correlation between the P slices is reduced. The areas of a frame that consist of macroblocks in SKIP mode will have a big change between the two P frames. Our skip mode detection technique mentioned in the previous section 4.3 (page 56-58) will not be suitable for the main profile, so in this new framework for fast mode decision will not adopt this technique.

Having adopted the conditions for SKIP mode decision in P and B slices [55, 56] as the initial part of our proposed scheme, we are left with the task of predicting the respective modes of the remaining macroblocks while trying to avoid as much computation (for the RD cost evaluation of different modes) as possible. Its worth noting that the remaining macroblocks still contain a percentage of skipped macroblocks that were not classified with the above conditions for SKIP mode decision. Table 5-2 shows that H.264/AVC standard always can find more SKIP mode macorblocks than SKIP mode detection conditions in [55, 56].

Sequences	[55, 56]	H. 264
News	61%	73%
Silent	48%	64%
Container	55%	78%
Foreman	10%	21%
Car-phone	17%	32%

Table 5-2 Percentage of SKIP modes using the SKIP mode conditions for [55, 56] and H.264 (100 Slices, Q CIF, Full search used for RD evaluation, QP28)

In the proposed scheme, we introduce two heuristics for predicting a small set of decidable modes, thus achieving significant computational savings due to the avoidance of exhaustive evaluation of all the RD costs. The set of decidable modes in our scheme includes two subsets, namely the subset {SKIP, MODE_16*16, MODE_16*8, MODE 8*16} which requires 4 RD evaluations and the subset {INTRA16*16, INTRA4*4} which requires 148 RD evaluations (9 INTRA4*4 modes times sixteen blocks plus four INTRA16*16 modes). The distinction between the intra and inter modes subsets (including the SKIP mode), is based on the work of [55] and considers the relation of the Average Boundary Error (ABE) with the Average Rate (AR). The selection of inter only modes (including SKIP) saves 164 RD evaluations, while the selection of intra modes after the best inter mode is found saves 16 RD evaluations. The proposed heuristics for the decidable modes are based on the relation between costs of specific modes for the current macroblock with average costs of modes in the previous and *previous of the same type* slices. Since the second heuristic (b)) depends on the first (a)), both are applicable in a sequential manner (first a) then b)) in the baseline and main profiles of the standard (P and B slices). The heuristics used are:

a) The J_{MODE_16*16} cost of the current macroblock should be less than the average J_{MODE_16*16} cost of the macroblocks in this mode *in the previous slice of the same slice type.*

b) For all macroblocks *not* satisfying heuristic a), the J_{INTER_8*8} cost of

the current macroblock should be less than the average J_{INTER_8*8} cost of the macroblocks *in the previous slice that are neither skipped nor were satisfied with heuristic a).* Furthermore, the J_{MODE_16*16} cost should be less than J_{INTER_8*8} for the current macroblock.

For the first heuristic, we need to stress two points. Firstly, that the RD cost of the MODE_16*16 is known (for the P slice macroblocks) from the SKIP mode decision part of our scheme, so no extra computation is needed. A single RD evaluation is obviously needed for B slice macroblocks and for both slice types a simple mean cost calculation is also required. Secondly, temporal rather than spatial information was chosen for designing this adaptive cost inequality, since although MODE_16*16 is highly likely for smooth areas in the current slice [67, 70], it is the temporal rather than the spatial information that is more relevant to the J costs. For the second heuristic, we only need to perform a single RD cost evaluation for 8x8 block size for both slice types, plus to find the average of the J_{INTER_8*8} cost for the macroblocks of the previous slice that were neither skipped nor caught by heuristic a). Since the J_{INTER_8*8} costs for these macroblocks are already known from the application of the second heuristic in the previous slice, the increase in the computational complexity due to the average calculation is negligible. The last point regarding our second heuristic is that since the set of decidable modes in our scheme contains inter modes of large block sizes and given the fact that the J_{MODE_16*16} and J_{INTER_8*8} costs are already known, it is reasonable to compare the J_{MODE_16*16} cost of the decidable modes with the cost of the first inter mode not belonging to our decidable set (J_{INTER_8*8}) We found experimentally that there is a high probability that a macroblock has a decidable mode, if the cost of the larger block size (16x16) is less than the cost of the smaller block size (8x8).

The choice of the actual modes included in the decidable set stems from our observation that while groupings of inter modes of large blocks (including intra modes) can be effectively achieved through the use of proper heuristics, the direct assignment of specific modes to macroblocks is unlikely to be achieved with computationally cheap techniques. The concept of grouping modes together was also proposed in [61, 77, 95]

but with different mode groups. For the macroblocks that are neither skipped nor they belong to the set of predictable modes, we finally evaluate the J_{INTER_4*4} cost and we use the monotonicity property [77] of the cost functions. According to this property, if $J_{MODE\_16*16} < J_{INTER\_8*8} < J_{INTER\_4*4}$ or $J_{MODE\_16*16} > J_{INTER\_8*8} > J_{INTER\_4*4}$, we only need to examine a subset of inter modes plus we need to distinguish between inter and intra modes. For the first inequalities (minimum cost is J_{MODE_16*16}), we evaluate the RD costs of the modes in the set {SKIP,MODE_16*16,MODE_16*8,MODE_8*16}, while for the second inequalities (minimum cost is J_{INTER_4*4}) we examine the modes in the set {SKIP,MODE_8*8,MODE_4*8,MODE_8*4,MODE_4*4}. The distinction between inter and intra modes is performed in exactly the same manner as in the case of determining decidable modes above. The reduction in the cardinality of the set of the inter modes to be considered, results after the selection of inter/intra modes in savings of 164 RD evaluations (intra case) and 16 RD evaluations (inter case) for the first inequalities. For the second inequalities, the savings are 163 and 15 RD evaluations for the intra and inter cases respectively. There are two interesting points to note regarding the design of our scheme at this stage. Firstly, the SKIP mode is still included in the set of inter modes to be evaluated, in order to catch any skipped macroblocks missed by Jeon's conditions [55, 56] and due to the fact that the SKIP mode RD evaluation is computationally cheap (no motion estimation). Secondly, the RD evaluation of the MODE_8*8 is still included for the second inequalities, despite the INTER_8*8 mode having already been examined in our second heuristic above. This is due to the fact that the cost J_{MODE_8*8} in this stage depends on the RD evaluation of smaller size sub-blocks, thus being potentially different to the J_{INTER_8*8} cost used above. The concept of monotonicity was also proposed in [77], but that work used thresholds for SKIP mode detection, gave results for the baseline profile only and attempted to gain computational savings through the simultaneous application of fast mode decision combined with a fast sub-pixel based motion estimation technique. In this work, we use the full motion estimation and the fast motion estimation techniques as adopted in the standard. Finally, our scheme concludes by performing exhaustive RD evaluation for any macroblocks not caught in the layers above. The proposed multi-layer mode decision scheme for baseline and main profiles can be summarized with the following

pseudo-code and block diagram:

BEGIN

1. If (all conditions for SKIP mode decision in P or B slices are satisfied)

{ $MODE_{CURRENT\_MACROBLOCK} = SKIP$; return} else go to Step 2

2. If (the J_{MODE_16*16} cost of the current macroblock is less than the average J_{MODE_16*16} cost of the macroblocks in this mode *in the previous slice of the same slice type*)

{ $INTER\_MODE_{CURRENT\_MACROBLOCK} \in \{SKIP, MODE\_16*16, MODE\_16*8, MODE\_8*16\}$

Perform INTRA/INTER mode decision; return} else go to Step 3

3. If (the J_{INTER_8*8} cost of the current macroblock is less than the average J_{INTER_8*8} cost of the macroblocks *in the previous slice that are neither skipped nor have they been caught by Step 2* AND the J_{MODE_16*16} cost is less than J_{INTER_8*8} for the current macroblock)

{ $INTER\_MODE_{CURRENT\_MACROBLOCK} \in \{SKIP, MODE\_16*16, MODE\_16*8, MODE\_8*16\}$ P erform INTRA/INTER mode decision; return} else go to Step 4

4. If (the monotonicity relation $J_{MODE\_16*16} < J_{INTER\_8*8} < J_{INTER\_4*4}$ holds true)

{ $INTER\_MODE_{CURRENT\_MACROBLOCK} \in \{SKIP, MODE\_16*16, MODE\_16*8, MODE\_8*16\}$ P erform INTRA/INTER mode decision; return }

else if (the monotonicity relation $J_{MODE\_16*16} > J_{INTER\_8*8} > J_{INTER\_4*4}$ holds true)

{ $INTER\_MODE_{CURRENT\_MACROBLOCK} \in \{SKIP, MODE\_8*8, MODE\_8*4, MODE\_4*8, MODE\_4*4\}$

Perform INTRA/INTER mode decision; return; }

For the rest of the macroblocks all modes are checked.

END

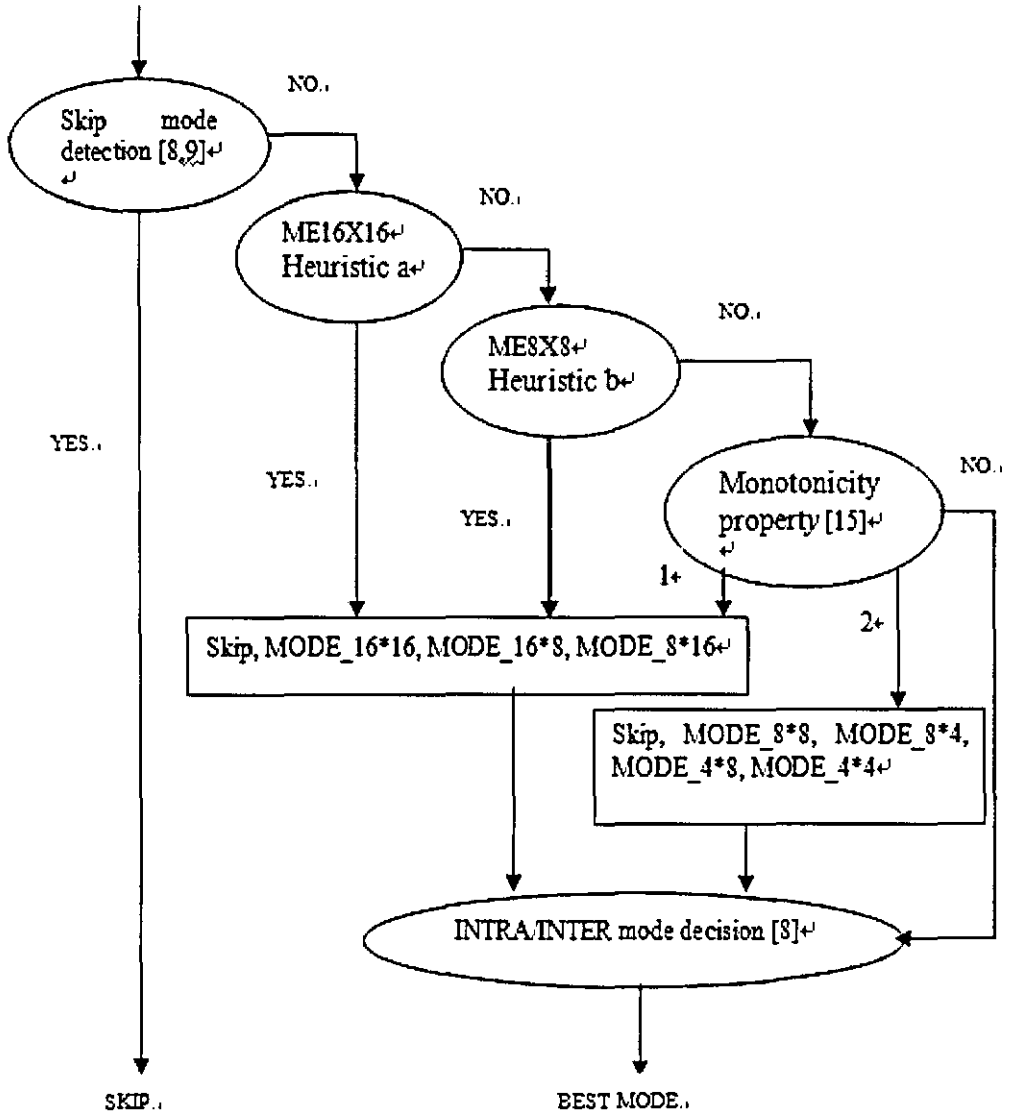


Figure 5-1 Block Diagram of Algorithm

The benefits of choosing *the spatial average costs of 16x16 and 8x8 macroblocks* for designing heuristics for mode decision can also be demonstrated statistically, as shown in figure 4-6 and 4-7 below. After all macroblocks in a slice have been encoded, these graphs show the conditional probability of correct mode decision, given these moving average costs as adaptive thresholds. It is worth noting that th1 and th2 correspond to the thresholds for the two different size blocks used for mode decision in steps 2 and 3 of the above pseudo-code. Three sequences with distinct motion characteristics (fast, medium, slow) were examined for the baseline and main profiles and each sequence consisted of a hundred slices in QCIF. In figure 5-2 and 5-3, the three lighter lines depict the fixed quantisation parameter and full search scenario (QP=28), while the three darker lines show the rate controlled and fast search scenario (target rate is 45 kbits). It can be clearly seen that by using the above heuristics, the probability of wrong prediction per slice is lower than 19 and 16 % for the simple and main profiles respectively.

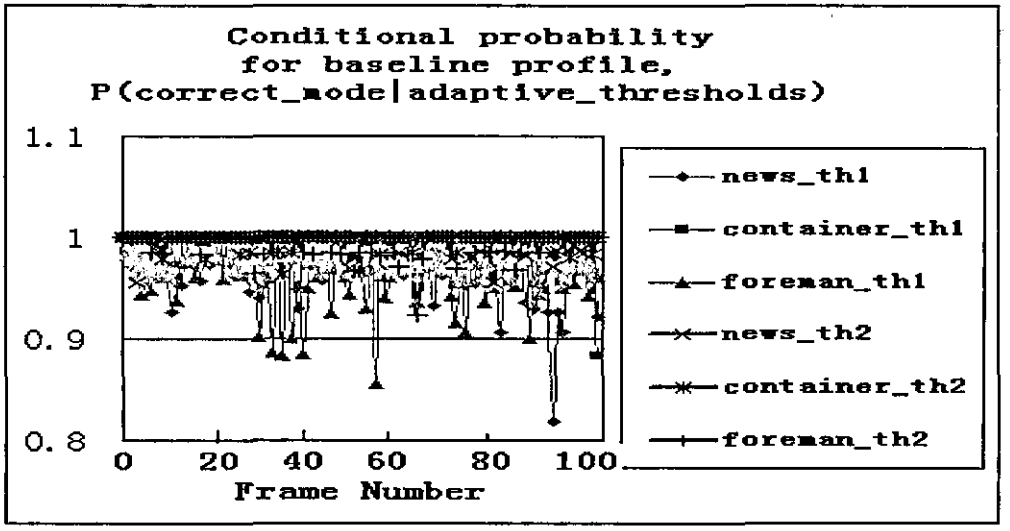


Figure 5-2

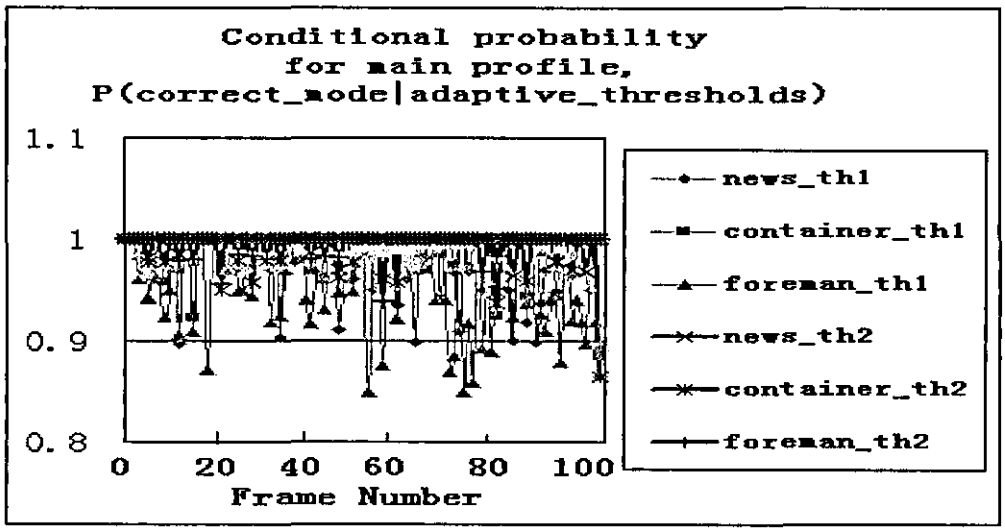


Figure 5-3

5.4 Experiments and Discussion

To assess the proposed scheme, a comprehensive set of experiments for a variety of video sequences with different motion characteristics was performed on a Pentium 4 (3.06 GHz processor). In particular, the exact characteristics of the test sequences are summarized in the following paragraph. Because a description of some of the test video sequences has already been mentioned in section 4.4 (page 63), here we give a description of some video sequences not included in the previous experiment.

"Tempete" is a sequence of leaves falling and plants blown, spatial detail, fast random motion and camera zoom. "Mobile" contains slow panning, zooming, a complex combination of horizontal motion like train, vertical motion like calendar, rotational motion like ball and high spatial colour detail. "Stefan" contains panning motion and has distinct fast changes in motion. "Coastguard" contains dominant horizontal motion of objects (two vessels) moving with different speed and different directions, oscillating motion (water motion) and camera panning, while "Flowergarden" is a slow "drive-by" of a house and its garden, containing camera panning and its dominant fast horizontal motion.

Both the cases of rate controlled and non rate controlled mode prediction were

examined in the Joint Model 8.2 (JM model) of the standard. In the set of metrics, we included both run times (in seconds) and a representative set of cycle counts from a static analysis of the Pentium 4 processor. It has to be noted that the static analysis does not measure run time program behaviour that benefits from cache locality and thus the gains are expected to be different as compared to run times. It is a useful metric mainly for embedded processor designers. Furthermore, according to our observations run times are more meaningful from the user's point of view. We compared our results with the standard and with the algorithms of Jeon et. al [55, 56] ,which was proposed as input to the standard and is based on the set of conditions for SKIP mode decision. Comparisons were made in terms of percentage rate differences, distortion differences, run time percentage differences (in seconds) and cycle percentage differences with respect to Joint Model 8.2 (JM model). The accuracy of the first and second heuristics for a variety of encoder configurations is shown in Tables 5-3 and 5-4 and the accuracy of the monotonicity conditions is shown in Table 5-5. The percentage accuracy in these tables denotes the percentage ratio of macroblocks with correctly predicted modes (as in H.264) over the total number of macroblocks and was measured independently for each heuristic and for the monotonicity conditions. In all the tables and graphs presented in this paper, RC denotes rate control at a specific target rate, QP denotes a fixed quantisation parameter, FS denotes full search and FME denotes fast motion estimation. In Tables 5-3 to 5-5, M and B also denote main and baseline profiles respectively. We tested the different components of our scheme (in terms of the metrics mentioned above) in an incremental fashion in graphs 5-1 to 5-21 and in Tables 5-6 to 5-15. This test methodology was chosen intentionally for *providing more insight* regarding the relative merits of the proposed components and the associated interactions between them. In graphs 5-1 to 5-21, the numbers {1,2,3,4} on the horizontal axis of the graphs denote the algorithms of Jeon [55, 56] (number 1), of Jeon plus monotonicity conditions (number two), of Jeon plus the first heuristic plus monotonicity conditions (number three) and of Jeon plus the two heuristics plus monotonicity conditions (number four). In tables 5-6 to 5-15, the Bjontegaard Delta Bit Rate (BDBR) and the Bjontegaard Delta PSNR (BDPSNR) metrics [96] were used to asses the rate distortion performance, for a fair comparison with previously published work. Furthermore, the intra refresh period was set

to 5 for main profile in tables 5-14 and 5-15.

The chosen search range was 32 pixels for both the full and the fast motion estimations. The chosen quantisation parameters were 28, 32, 36, and 40 for CIF, QCIF and 720x480 sequences (100 slices, intra refresh period 105 slices). For rate control, target bit rates of 45 and 20 kbits were chosen for QCIF sequences, and 500 and 250 kbits for CIF sequences and 2Megabits/S for 720x480 sequences. The configuration file for the encoder had the following settings: Level 30, RD optimization ON, Hadamard Transform, IPPP... and IBBP structure, CABAC coding, no error tools, number of reference slices was 5. For rate control, the initial QP was 24.

Sequences	QP 28, FS,B	QP 28, FME,B	RC45k FS,S	RC45k FME,B	QP 28, FS,M	QP 28, FME,M	RC45k FS,M	RC45k FME,M
News	97%	97%	99%	99%	96%	95%	98%	98%
Silent	96%	96%	99%	99%	91%	91%	98%	98%
Container	98%	98%	97%	98%	99%	98%	97%	99.5%
Foreman	96%	96%	99%	99%	94%	94%	99.7%	99.5%
Carphone	98%	97%	99%	99%	94%	93%	98%	98%

Table 5-3 Percentage Accuracy of the First Heuristic

Sequences	QP28 FS,B	QP28, FME,B	RC45k FS,B	RC45k FME,B	QP28, FS,M	QP 28, FME,M	RC45k FS,M	RC45k FME,M
News	99%	99%	99%	99%	95%	98%	99.7%	99.7%
Silent	99%	98%	99.7%	99.7%	97%	98%	99.7%	99.7%
Container	99.8	99.8%	99.7%	99.7%	99%	97%	99%	98%
Foreman	97%	98%	99.5%	99%	94%	94%	99.5%	99%
Carphone	98%	98%	99%	99%	98%	97%	99%	99%

Table 5-4 Percentage Accuracy of the Second Heuristic

Sequences	QP28 FS,B	QP 28, FME,B	RC45k FS,B	RC45k FME,B	QP 28, FS,M	QP 28, FME,M	RC45k FS,M	RC45k FME,M
News	97%	96%	98%	98%	93%	94%	98%	98%
Silent	97%	96%	98%	98%	93%	93%	97%	97%
Container	98%	98%	98%	98%	98%	98%	97%	96%
Foreman	93%	94%	99%	99%	87%	86%	98%	97%
Carphone	95%	94%	98%	98%	94%	93%	98%	97%

Table 5-5 Percentage Accuracy of Monotonicity Conditions

Video sequences	Res	Profile	ME	JEON			JEON+MON		
				BDPSNR(dB)	*BDBR (%)*	*TS (%)*	*BDPSNR (dB)*	*BDBR (%)*	*TS (%)*
news	qcif	base	FS	-0.032	-0.23	46.95	-0.066	0.055	52.25
silent	qcif	base	FS	-0.016	0	37.76	-0.029	0.385	45.38
container	qcif	base	FS	0.059	-1.355	51.55	0.026	-0.813	57.41
foreman	qcif	base	FS	-0.012	-0.2	17.65	-0.023	-0.41	31.11
carphone	qcif	base	FS	-0.069	0.153	14.90	-0.079	-0.315	32.63
news	qcif	main	FS	0.004	-1.35	62.58	-0.080	-0.305	67.49
silent	qcif	main	FS	-0.013	-0.283	55.74	-0.077	0.653	62.29
container	qcif	main	FS	0.056	-1.753	73.42	0.057	-2.248	76.64
foreman	qcif	main	FS	-0.054	-2.138	42.60	-0.110	-1.595	48.62
carphone	qcif	main	FS	-0.063	-2.633	43.52	-0.156	-2.228	52.79

Table 5-6 BDBR/BDPSNR/TIME SAVINGS performance of Jeon's and Jeon's+monotonicity schemes

Video sequences	Res	Profile	ME	JEON+ONE+MON			JEON+ONE+TWO+MON		
				BDPSNR (dB)	*BDBR (%)*	*TS (%)*	*BDPSNR (dB)*	*BDBR (%)*	*TS (%)*
news	qcif	base	FS	-0.055	-0.308	56.72	-0.081	0.138	59.77
silent	qcif	base	FS	-0.022	0.07	48.85	-0.024	0.13	52.96
container	qcif	base	FS	0.025	-0.743	61.49	0.008	-0.63	64.31
foreman	qcif	base	FS	-0.035	-0.24	37.84	-0.053	-0.28	43.23
carphone	qcif	base	FS	-0.098	-0.22	39.73	-0.11	-0.04	44.36
news	qcif	main	FS	-0.081	-0.48	70.45	-0.086	-0.465	71.68
silent	qcif	main	FS	-0.081	0.895	65.77	-0.082	0.925	66.99
container	qcif	main	FS	0.034	-1.78	79.3	0.056	-2.073	79.97
foreman	qcif	main	FS	-0.14	-1.388	56.31	-0.114	-1.67	57.47
carphon	qcif	main	FS	-0.125	-1.078	58.48	-0.122	-1.99	59.46

Table 5-7 BDBR/BDPSNR/TIME SAVINGS performance of the proposed scheme(ONE and TWO are steps 2 and 3 in the pseudocode)

Video sequences	Res	Profile	ME	JEON			JEON+MON		
				BDPSNR (dB)	*BDBR (%)*	*TS (%)*	*BDPSNR (dB)*	*BDBR (%)*	*TS (%)*
news	qcif	base	FME	-0.029	-0.18	64.11	-0.058	-0.135	67.81
silent	qcif	base	FME	-0.04	-0.018	55.23	-0.037	-0.145	60.91
container	qcif	base	FME	0.068	-1.483	70.63	0.015	-0.665	74.22
foreman	qcif	base	FME	-0.024	-0.143	37.8	-0.079	-0.222	47.97
carphone	qcif	base	FME	-0.044	-0.268	43.51	-0.07	-0.113	53.15
news	qcif	main	FME	-0.009	-0.988	76.12	-0.072	-0.59	78.32
silent	qcif	main	FME	-0.032	-0.243	70.6	-0.06	0.135	73.38
container	qcif	main	FME	0.067	-1.76	84.18	0.085	-2.388	85.55
foreman	qcif	main	FME	-0.054	-1.885	58.56	-0.085	-2.238	60.58
carphone	qcif	main	FME	-0.073	-2.055	60.81	-0.121	-2.648	66.07

Table 5-8 BDBR/BDPSNR/TIME SAVINGS performance of Jeon's and Jeon's+monotonicity schemes

Video sequences	Res	Profile	ME	JEON+ONE+MON			JEON+ONE+TWO+MON		
				BDPSNR (dB)	*BDBR (%)*	*TS (%)*	*BDPSNR (dB)*	*BDBR (%)*	*TS (%)*
news	qcif	base	FME	-0.086	0.418	72.20	-0.088	0.348	73.24
silent	qcif	base	FME	-0.050	0.243	66.49	-0.041	0.183	67.71
container	qcif	base	FME	0.015	-0.705	78.21	0.015	-0.738	79.30
foreman	qcif	base	FME	-0.082	0.035	56.15	-0.087	-0.083	58.22
carphone	qcif	base	FME	-0.090	-0.05	59.92	-0.100	-0.123	61.39
news	qcif	main	FME	-0.081	-0.45	79.68	-0.081	-0.388	80.68
silent	qcif	main	FME	-0.094	0.513	75.97	-0.081	0.26	76.98
container	qcif	main	FME	0.086	-2.518	87.02	0.074	-2.138	87.77
foreman	qcif	main	FME	-0.079	-2.36	66.36	-0.082	-2.223	67.76
carphone	qcif	main	FME	-0.146	-2.17	70.58	-0.152	-2.14	71.88

Table 5-9 BDBR/BDPSNR/TIME SAVINGS performance of the proposed scheme(ONE and TWO are steps 2 and 3 in the pseudocode

Video sequences	Res	Profile	ME	JEON			JEON+MON		
				BDPSNR (dB)	*BDBR (%)*	*TS (%)*	*BDPSNR (dB)*	*BDBR (%)*	*TS (%)*
foreman	Cif	base	FS	-0.061	0.138	19.39	-0.098	0.22	27.66
coastguard	Cif	base	FS	-0.011	-0.835	14.30	-0.023	-1.55	23.63
stefan	Cif	base	FS	-0.036	-0.345	14.70	-0.107	0.83	23.89
mobile	Cif	base	FS	-0.001	-0.403	10.32	-0.021	-0.63	20.74
tempete	Cif	base	FS	-0.011	-0.303	13.67	-0.021	-0.633	22.42
foreman	Cif	main	FS	-0.116	-0.913	40.73	-0.188	-0.195	42.87
coastguard	Cif	main	FS	-0.043	-2.178	34.88	-0.062	-2.265	41.82
stefan	Cif	main	FS	-0.052	-0.718	28.21	-0.203	1.695	36.67
mobile	Cif	main	FS	-0.044	-0.324	25.52	-0.117	0.233	34.84
tempete	Cif	main	FS	-0.054	-0.995	32.58	-0.090	-0.913	38.41

Table 5-10 BDBR/BDPSNR/TIME SAVINGS performance of Jeon's and Jeon's+monotonicity schemes

Video sequences	Res	Profile	ME	JEON+ONE+MON			JEON+ONE+TWO+MON		
				BDPSNR (dB)	*BDBR (%)*	*TS (%)*	*BDPSNR (dB)*	*BDBR (%)*	*TS (%)*
foreman	Cif	base	FS	-0.100	0.37	34.85	-0.103	0.408	39.85
coastguard	Cif	base	FS	-0.025	-1.525	31.04	-0.026	-1.53	37.02
stefan	Cif	base	FS	-0.118	0.74	29.05	-0.125	0.763	35.67
mobile	Cif	base	FS	-0.033	-0.62	27.56	-0.027	-0.823	33.70
tempete	Cif	base	FS	-0.028	-0.688	29.63	-0.023	-0.715	35.05
foreman	Cif	main	FS	-0.192	-0.035	51.31	-0.187	-0.173	52.61
coastguard	Cif	main	FS	-0.065	-2.52	47.86	-0.059	-2.805	49.51
stefan	Cif	main	FS	-0.233	2.333	45.01	-0.237	2.255	46.45
mobile	Cif	main	FS	-0.128	0.14	44.74	-0.133	0.275	45.45
tempete	Cif	main	FS	-0.094	-0.975	46.37	-0.090	-1.033	47.74

Table 5-11 BDBR/BDPSNR/TIME SAVINGS performance of the proposed scheme(ONE and TWO are steps 2 and 3 in the pseudocode)

Video sequences	Res	Profile	ME	JEON			JEON+MON		
				BDPSNR (dB)	*BDBR (%)*	*TS (%)*	*BDPSNR (dB)*	*BDBR (%)*	*TS (%)*
foreman	Cif	base	FME	-0.076	0.013	41.07	-0.107	0.215	50.68
coastguard	Cif	base	FME	-0.040	-0.635	30.24	-0.037	-1.383	41.14
stefan	Cif	base	FME	-0.026	-0.503	38.48	-0.102	0.49	46.27
mobile	Cif	base	FME	-0.009	-0.405	32.98	-0.030	-0.69	41.34
tempete	Cif	base	FME	-0.020	-0.13	35.54	-0.037	-0.248	45.35
foreman	Cif	main	FME	-0.139	-0.558	57.39	-0.177	-0.12	60.13
coastguard	Cif	main	FME	-0.041	-2.875	48.57	-0.071	-3.108	54.73
stefan	Cif	main	FME	-0.065	-0.623	47.47	-0.214	0.663	52.95
mobile	Cif	main	FME	-0.045	-0.51	43.76	-0.123	0.493	49.62
tempete	Cif	main	FME	-0.055	-1.018	50.77	-0.093	-0.88	57.10

Table 5-12 BDBR/BDPSNR/TIME SAVINGS performance of Jeon's and Jeon's+monotonicity schemes

Video sequences	Res	Profile	ME	JEON+ONE+MON			JEON+ONE+TWO+MON		
				BDPSNR (dB)	*BDBR (%)*	*TS (%)*	*BDPSNR (dB)*	*BDBR (%)*	*TS (%)*
foreman	Cif	base	FME	-0.109	0.18	58.24	-0.113	0.298	59.99
coastguard	Cif	base	FME	-0.031	-1.473	49.51	-0.032	-1.673	51.35
stefan	Cif	base	FME	-0.106	0.32	53.28	-0.105	0.523	55.4
mobile	Cif	base	FME	-0.036	-0.673	49.18	-0.034	-0.668	51.09
tempete	Cif	base	FME	-0.031	-0.353	53.24	-0.033	-0.398	55.32
foreman	Cif	main	FME	-0.191	-0.18	65.29	-0.185	-0.27	66.48
coastguard	Cif	main	FME	-0.058	-3.3	59.98	-0.061	-3.173	61.22
stefan	Cif	main	FME	-0.229	1.888	59.73	-0.225	1.955	60.54
mobile	Cif	main	FME	-0.127	0.258	56.41	-0.139	0.273	57.68
tempete	Cif	main	FME	-0.103	-0.68	61.40	-0.112	-0.663	62.57

Table 5-13 BDBR/BDPSNR/TIME SAVINGS performance of the proposed scheme(ONE and TWO are steps 2 and 3 in the pseudocode)

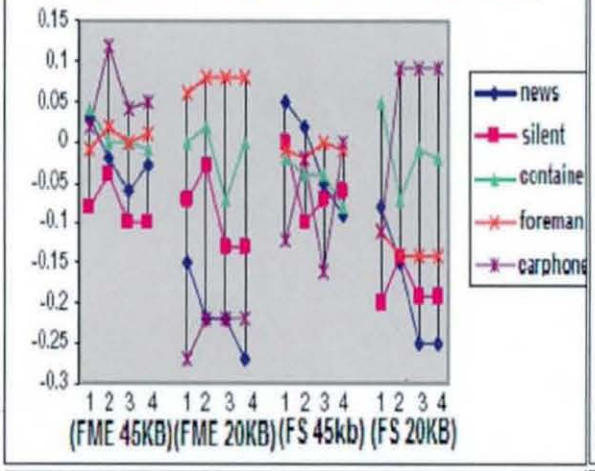
Video sequences	Res	Profile	ME	JEON			JEON+MON		
				BDPSNR (dB)	*BDBR (%)*	*TS (%)*	*BDPSNR (dB)*	*BDBR(%)*	*TS(%)*
mobile	720x480	base	FME	-0.018	-0.185	23.87	-0.047	-0.533	30.98
tempete	720x480	base	FME	-0.049	-0.458	31.85	-0.057	-1.038	40.99
flower	720x480	base	FME	-0.024	-0.275	34.75	-0.060	-0.605	39.82
stefan	720x480	base	FME	-0.065	-0.495	32.12	-0.077	-1.22	39.5
mobile	720x480	refresh	FME	-0.016	-0.23	33.85	-0.080	0.118	42.79
tempete	720x480	refresh	FME	-0.024	-0.465	47.28	-0.047	-0.755	53.65
flower	720x480	refresh	FME	-0.021	-0.158	41.09	-0.084	0.23	46.18
stefan	720x480	refresh	FME	-0.029	-0.89	40.53	-0.062	-0.555	49.25
mobile	720x480	main	FME	-0.034	-0.21	33.42	-0.095	-0.068	42.79
tempete	720x480	main	FME	-0.036	-0.723	46.1	-0.062	-1.363	53.11
flower	720x480	main	FME	-0.023	-0.143	40.48	-0.091	0.193	45.83
stefan	720x480	main	FME	-0.032	-1.258	40.41	-0.095	-1.178	49.28

Table 5-14 BDBR/BDPSNR/TIME SAVINGS performance of Jeon's and Jeon's+monotonicity schemes

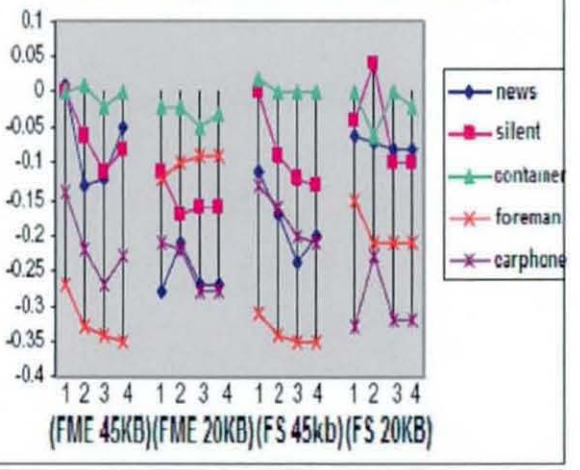
Video sequence	Res	Profile	ME	JEON+ONE+MON			JEON+ONE+TWO+MON		
				BDPSNR (dB)	*BDBR (%)*	*TS (%)*	*BDPSNR (dB)*	*BDBR(%)*	*TS(%)*
mobile	720x480	base	FME	-0.050	-0.655	39.17	-0.056	-0.678	41.15
tempete	720x480	base	FME	-0.053	-1.165	49.97	-0.056	-1.108	51.56
flower	720x480	base	FME	-0.064	-0.583	46.25	-0.072	-0.54	47.83
stefan	720x480	base	FME	-0.084	-2.835	47.35	-0.080	-1.223	48.89
mobile	720x480	refresh	FME	-0.082	0.003	46.3	-0.082	0.085	47.76
tempete	720x480	refresh	FME	-0.047	-0.785	57.71	-0.048	-0.783	58.84
flower	720x480	refresh	FME	-0.091	0.33	49.15	-0.092	0.225	50.18
stefan	720x480	refresh	FME	-0.064	-0.875	51.74	-0.051	-0.903	53.37
mobile	720x480	main	FME	-0.103	-0.153	47.78	-0.106	0.05	48.83
tempete	720x480	main	FME	-0.067	-1.415	58.16	-0.067	-1.4	58.99
flower	720x480	main	FME	-0.102	0.063	50.37	-0.11	0.25	51.09
stefan	720x480	main	FME	-0.066	-2.383	53.86	-0.085	-1.73	54.68

Table 5-15 BDBR/BDPSNR/TIME SAVINGS performance of the proposed schemes(ONE and TWO are steps 2 and 3 in the pseudocode)

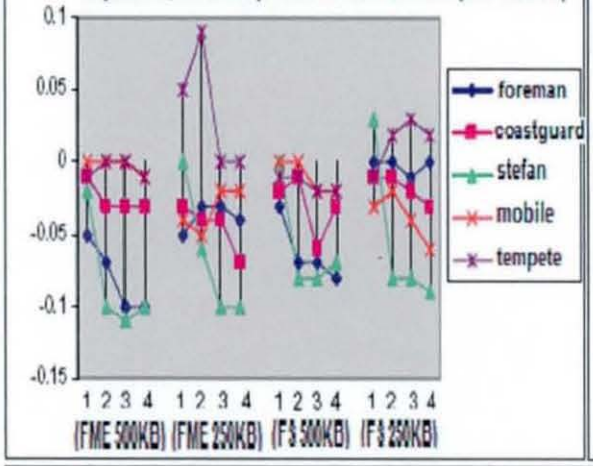
PSNR Graph 5-1 PSNR differences for the four algorithms in qcif sequences, baseline profile for full and FME (rate control)



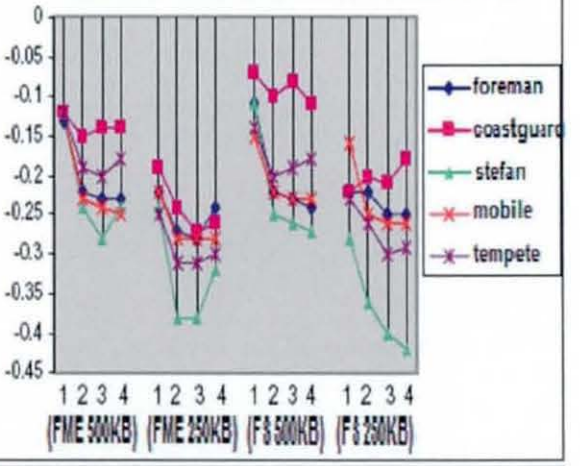
PSNR Graph 5-2 PSNR differences for the four algorithms in qcif sequences, main profile for full and FME (rateo control)



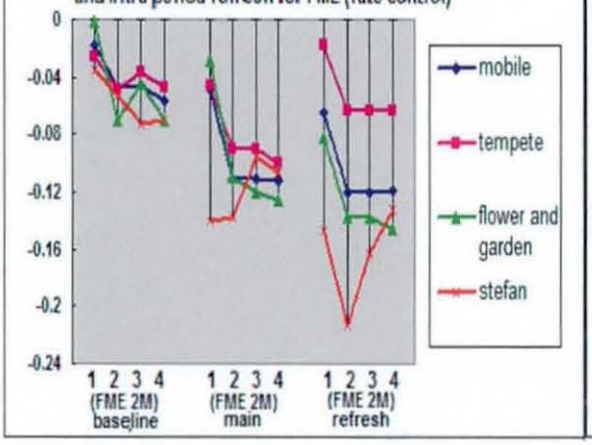
PSNR Graph 5-3 PSNR differences for the four algorithms in cif sequences, baseline profile for full and FME (rate control)



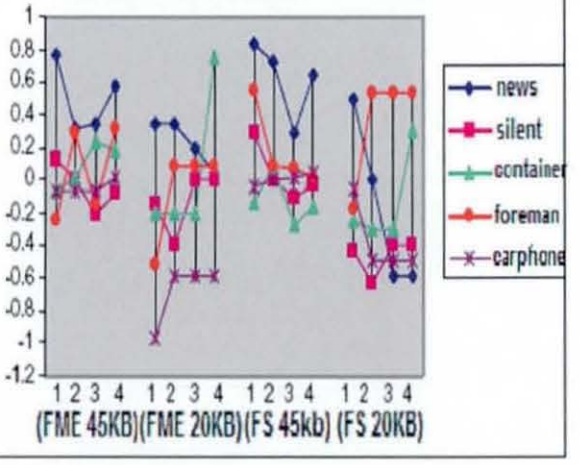
PSNR Graph 5-4 PSNR differences for the four algorithms in cif sequences, main profile for full and FME (rate control)



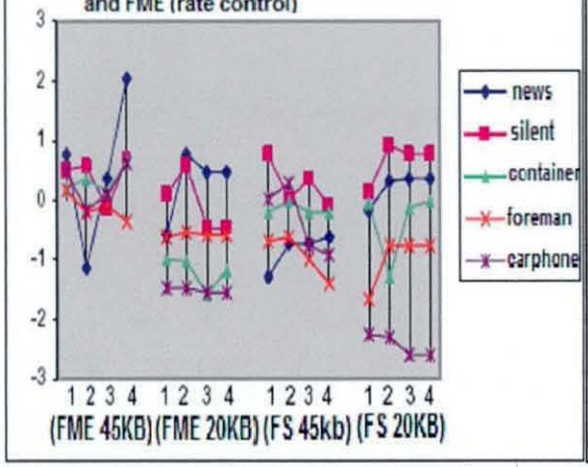
PSNR Graph 5-5 PSNR differences for the four algorithms in 720x480 sequences, baseline、main profiles and intra period refresh for FME (rate control)



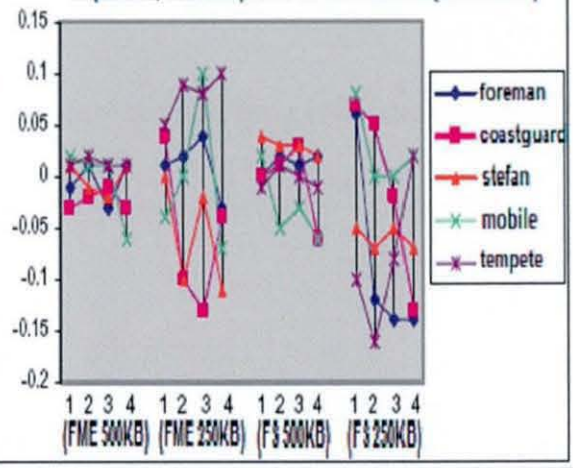
bitrate% Graph 5-6 bitrate %differences for the four algorithms in qcif sequences, baseline profile for full and FME (rate control)



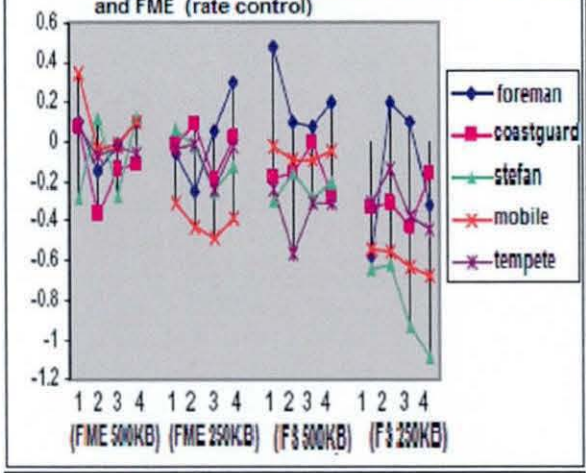
bitrate% Graph 5-7 bitrate %differences for the four algorithms in qcif sequences, main profile for full and FME (rate control)



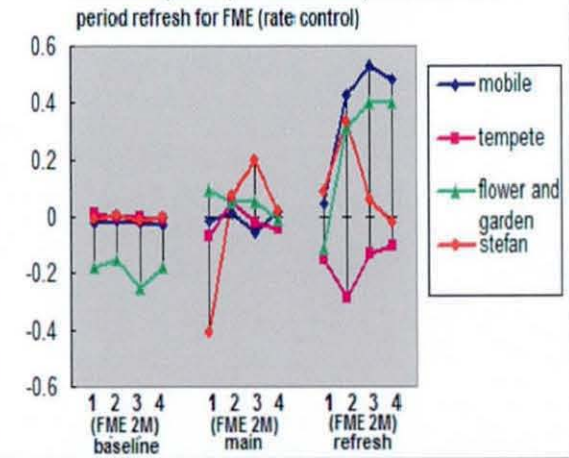
bitrate% Graph 5-8 bitrate % differneces for the four algorithms in cif sequences, baseline profile for full and FME (rate control)



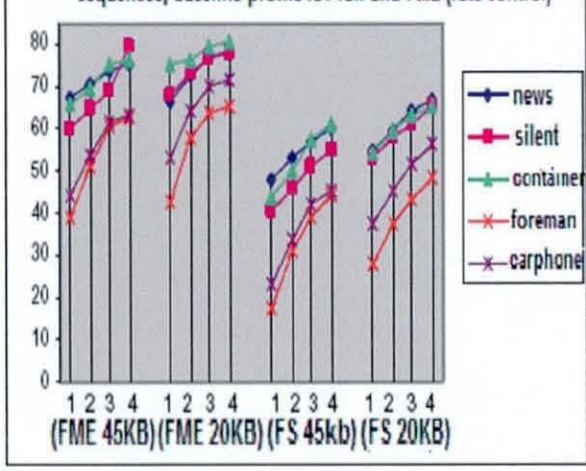
bitrate% Graph 5-9 bitrate %differences for the four algorithms in cif sequences, main profile for full and FME (rate control)



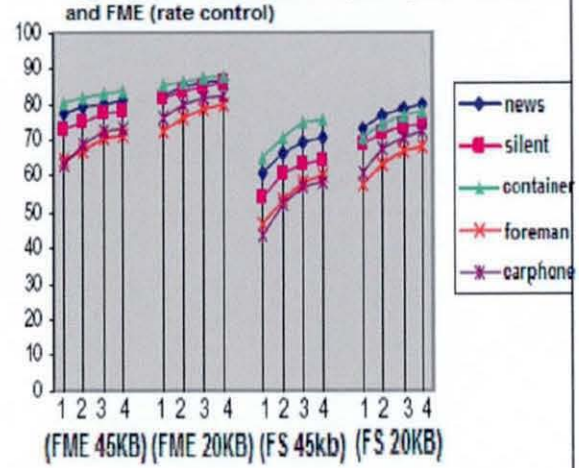
bitrate% Graph 5-10 bitrate %differences for the four algorithms in 720x480 sequences, baseline、main profiles and intra period refresh for FME (rate control)



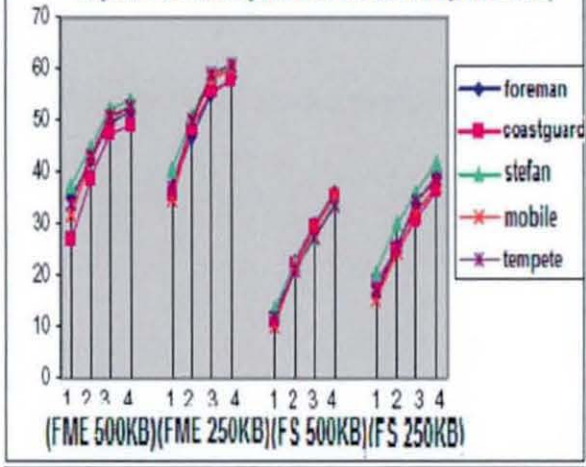
Time% Graph 5-11 Time %differences for the four algorithms in qcif sequences, baseline profile for full and FME (rate control)



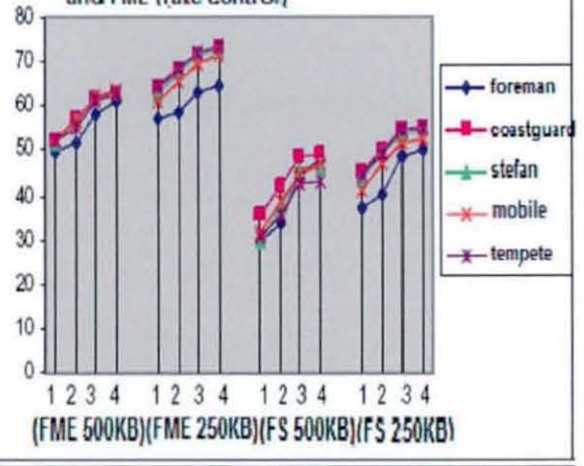
Time% Graph 5-12 Time %differences for the four algorithms in qcif sequences, main profile for full and FME (rate control)



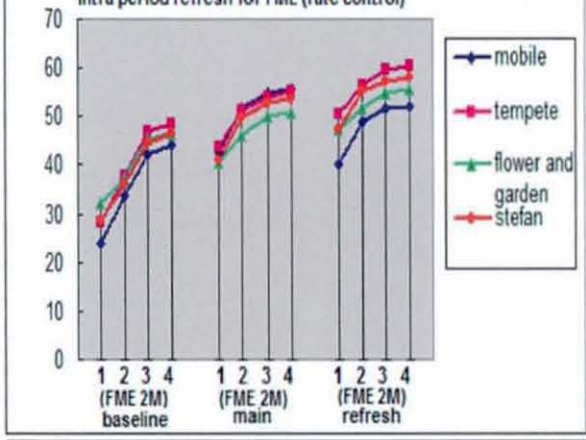
Time% Graph 5-13 Time %differences for the four algorithms in cif sequences, baseline profile for full and FME (rate control)



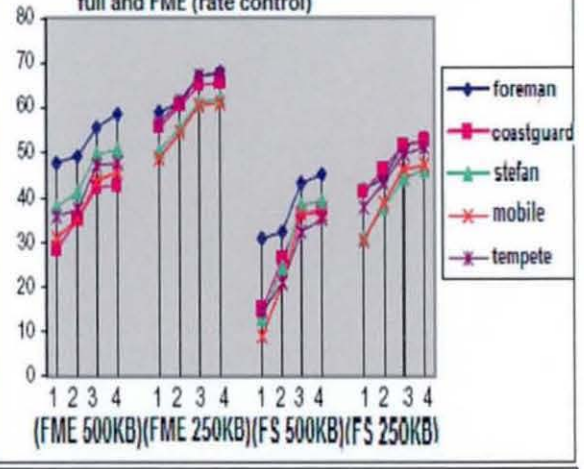
Time% Graph 5-14 Time %differences for the four algorithms in cif sequences, main profile for full and FME (rate control)



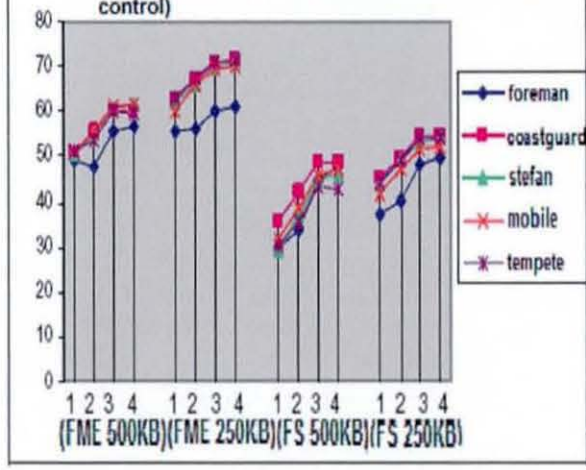
Time% Graph 5-15 Time %differences for the four algorithms in 720x480 sequences, baseline、main profiles and intra period refresh for FME (rate control)



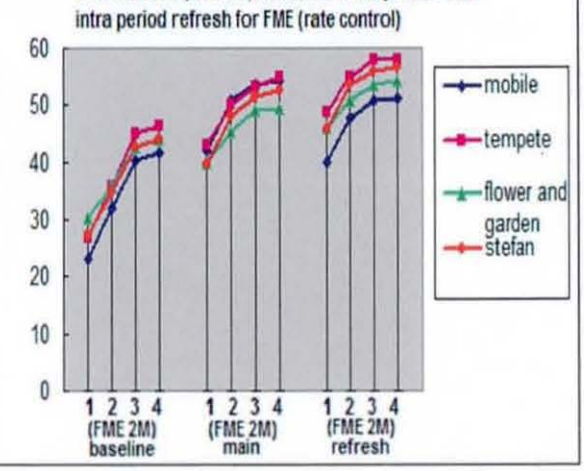
Cycles% Graph 5-16 Cycle %differences for the four algorithms in cif sequences, baseline profile for full and FME (rate control)



Cycles% Graph 5-17 Cycle %differences for the four algorithms in cif sequences, main profile for full and FME (rate control)

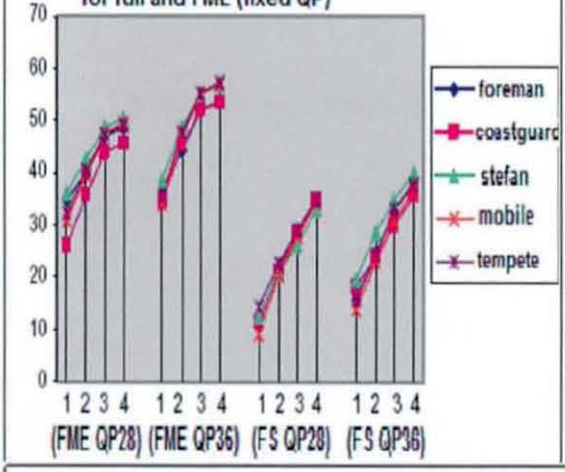


Cycles% Graph 5-18 Cycle %differences for the four algorithms in 720x480 sequences, baseline、main profiles and intra period refresh for FME (rate control)



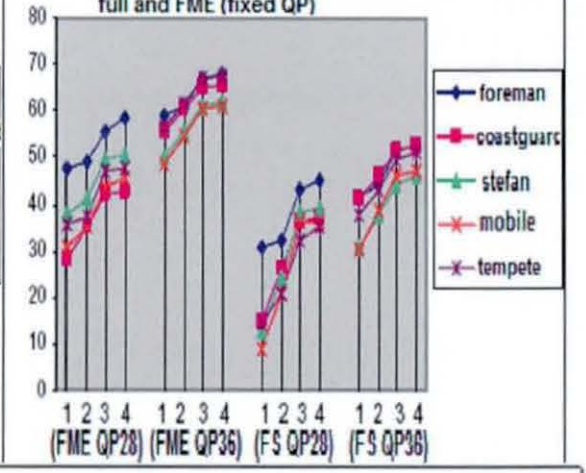
Cycle%

Graph 5.19 Cycle %differences for the four algorithms in cif sequences, baseline profile for full and FME (fixed QP)



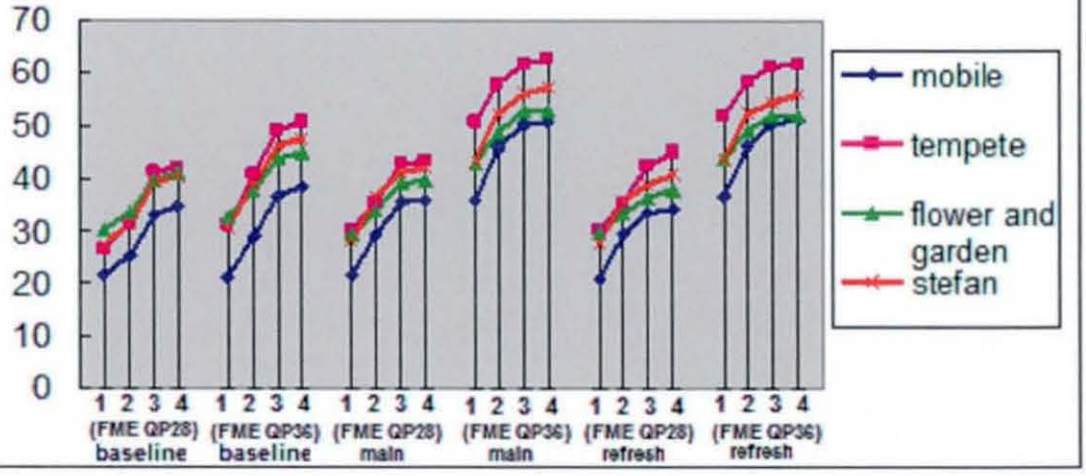
Cycle%

Graph 5.20 Cycle %differences for the four algorithms in cif sequences, main profile for full and FME (fixed QP)



Cycles%

Graph 5.21 Cycle %differences for the four algorithms in 720x480 sequences, baseline, main profiles and FME (fixed QP)



From the above tables and graphs, we could see that computational complexity reduction and quality reduction all depend on video motion characteristics. Our scheme can achieve more time savings and low quality reduction in slow motion sequences compared with fast motion ones. For example in table 5-7 we can gain an average of 64% time savings for Container (baseline profile, qcif format, FS) with BDPSNR increase 0.03dB and BDBR reduction 0.63%, but only an average of 43% time saving for Foreman (baseline profile, qcif format, FS) with BDPSNR reduction 0.035dB and BDBR reduction 0.28%. Computational complexity reduction also depends on quantization parameters and target bitrates. Our algorithm can attain more cycle savings at higher QP or lower target bitrates. Graph 5-19 shows that about 38% cycle savings can be achieved in Stefan at QP 36 (FS), but only about 30% cycle savings at QP28 (FS). We also can find similar trend in Graph 5-17. About 55% cycle savings can be achieved in Coastguard at 250k (FS), but only about 49% cycle savings at 500k (FS). For fixed quantisation parameters the bit rate percentage differences are in the range [-5.1, 1.5], while for the rate control case they are in the range [-8, 3]. In both cases the minus sign indicates percentage bit savings, while the plus sign, percentage of extra bits. The PSNR differences for the same encoder settings are in the range [-0.4, 0.12] with 0.2dB max difference as compared to the work in [55-57]. The minus sign in this case represents PSNR degradation, while the plus sign PSNR improvement. This clearly shows that the performance of the proposed scheme (either in its entirety or in terms of individual components) has very similar RD performance with H.264/AVC. The maximum PSNR degradation occurs for fast sequences with continuously changing motion such as "Stefan" and is expected due to the fact that neither early SKIP mode detection can be exploited effectively, nor can the stability of the cost functions in the previous slices due to the fast changing statistics in the video sequence. In terms of time savings, which are indeed video content dependent, we see percentage savings in the range [5.8-90.1], with increased percentage savings in the range 9-29% as compared to the work in [55-57]. The increasing time savings as we move from algorithm 1 to algorithm 4 in each of the graphs reveals that the addition of the individual components to the Jeon's work in [55-57] (such as monotonicity and the two heuristics) clearly benefits the time savings. Similar trends to time savings were observed for the percentage of cycles reduction (in selected encoder

settings) as shown in graphs 5-1 to 5-21. It can also be observed that for the same settings (fixed QPs or target bit rates), the algorithm is faster for fast motion estimation as compared to full search and this can also be expected since the time savings in this case are now due to the combination of fast mode prediction and the search method itself.

Finally, a comparison of our work with [60] (which outperforms the approaches in [30, 61]) reveals better RD performance, since average bit rate increases of 9.8% for similar average PSNR as the standard are reported.

5.5 Conclusion

We have proposed a fast mode decision scheme for the baseline and main profiles of the H.264/AVC standard that jointly exploits a set of conditions for SKIP mode decision, two heuristics that reduce the set of inter modes to be examined, inter/intra mode prediction and the monotonicity property of the cost functions. For RD performance very close to the standard, the proposed scheme affords significant time/cycle savings as compared to H.264/AVC and other known work that was proposed as input to the standard. Our scheme is very relevant to low complexity coding systems.

Chapter 6

Error Assisted Fast Mode Decision in H.264/AVC

6.1 Preamble

Two fast mode decision algorithms we propose in above two chapters use temporal information correlation to predict modes. Questions are arising: Is temporal information always more important than spatial information in fast mode prediction? Could we find some algorithms to joint the spatial and temporal information together? A new algorithm for fast mode decision [98] in the H.264/AVC video coding standard is presented in this chapter. The novelty of the algorithm stems from the fact that it jointly exploits error and pixel information for mode decision, as opposed to pixel only information [67, 70]. The exploitation of error information is achieved through the use of moving averages of error cost functions as decision steps for mode selection and results in significant computational savings with similar Rate Distortion performance as compared to [67, 70].

6.2 Previous Work

Because this new algorithm is an improvement over the algorithms of [67, 70], we need to explain [67, 70] firstly. In [67, 70], the intra/inter mode decision of the macroblock to be encoded is based on fixed thresholds, homogeneity and predicted motion characteristics. For the intra mode decision, an edge map is initially computed

using Sobel operators in the vertical/horizontal directions. Specifically, for every macroblock pixel $p_{i,j}$ an edge vector $\overrightarrow{E_{i,j}} = \{dx_{i,j}, dy_{i,j}\}$ is computed, where the $dx_{i,j}, dy_{i,j}$ components are local gradients:

$$dx_{i,j} = p_{i-1,j+1} + 2 * p_{i,j+1} + p_{i+1,j+1} - p_{i-1,j-1} - 2 * p_{i,j-1} - p_{i+1,j-1} \qquad 5\text{-}1$$

$$dy_{i,j} = p_{i+1,j-1} + 2 * p_{i+1,j} + p_{i+1,j+1} - p_{i-1,j-1} - 2 * p_{i-1,j} - p_{i-1,j+1} \qquad 5\text{-}2$$

The edge vector amplitude is $Amp(\overrightarrow{E_{i,j}}) = \sqrt{|dx_{i,j}| + |dy_{i,j}|}$. 5-3

And the direction is estimated by:

$$Ang(\overrightarrow{E_{i,j}}) = \frac{180^0}{\pi} * \arctan(\frac{dy_{i,j}}{dx_{i,j}}), \quad \left|Ang\left(\overrightarrow{D_{i,j}}\right)\right| < 90 \qquad 5\text{-}4$$

A histogram is then created by summing the edge vector amplitudes belonging to the same plane segment. Two such vectors $\overrightarrow{E_{i,j}}$ and $\overrightarrow{E_{k,l}}$ belong to the same segment if $A \le Ang(\overrightarrow{E_{i,j}}) \le B$ and $A \le Ang(\overrightarrow{E_{k,l}}) \le B$, with A and B segment limits. For the 4x4 intra mode case, the plane is split into eight segments (signifying eight modes), while for the 8x8 chroma and the 16x16 luma intra modes the plane is split into three segments, corresponding to horizontal, vertical and plane modes respectively. The total number of candidate modes in all cases is one more than the number of segments due to the inclusion of the DC mode. To reduce the computational cost, [70] only checks a subset of these intra modes. In the 4x4 case, only the mode of the histogram cell with the highest amplitude, the modes of its two adjacent cells and the DC mode are considered for a total of four (as opposed to nine) modes. In the 16x16 luma case, the mode of the cell with the highest amplitude along with the DC mode are checked, for a total of two (as opposed to four) modes. Finally, if the two histograms of the U and V chroma components have the same most probable mode, two modes are checked, otherwise three (including again the DC mode in both cases). In total, the number of Motion Estimations (MEs) and Rate Distortion Optimizations (RDOs) for intra mode prediction is reduced from 592 in

H.264/AVC to 132 or 198, depending on the number of checked modes for the chroma components.

Homogeneity is then assessed for inter mode prediction (both for 16x16 and 8x8 cases) by the sum of the amplitudes of the edge vectors. If this sum is less than a pre-determined threshold, the macroblock is homogeneous, otherwise it is not. Since in [70] histogram should be done for fast intra mode prediction, the only additional operations for determining homogeneity in inter mode (as compared to intra) are amplitude summations plus two extra comparisons with thresholds for the two block sizes. For the 16x16 case, if the dominant intra mode in the histogram is the horizontal one, only the SKIP, 16x16 and 16x8 modes are considered (no 8x8 mode checking is performed). Similarly, only the SKIP, 16x16 and 8x16 modes are considered in the case of a dominant vertical intra mode. Otherwise, only the 16x16 and SKIP modes are checked. The inclusion of smaller block sizes in the mode decision for 16x16 homogeneous macroblocks accounts for potentially bad predictions in the ME stage. For 8x8 homogeneous blocks, only the 8x8 mode is considered, thus saving RDOs and MEs for 8x4, 4x8 and 4x4 block sizes in P and B slices. Finally, if the Sum of Absolute Differences (SAD) between this block and its collocated one is less than a pre-specified threshold, the current macroblock has zero motion and the only modes checked are the 16x16 and the SKIP ones. In total, the number of MEs and RDOs for *inter* mode prediction of 16x16 block sizes is reduced from four to three or two corresponding to the cases of horizontal/vertical intra dominant modes and plane modes/ zero motion macroblock respectively. For the 8x8 block size, the reduction of ME and RDO evaluations is three. The above inter mode prediction scheme is illustrated with the following pseudocode and block diagram (figure 6-1):

1) Do edge detection on the macroblock to be encoded and generate the edge direction histogram

2) *Predict* if the current macroblock has zero motion. If the SAD between the current and collocated macroblocks is less than 200, only check SKIP and 16*16 modes. If not, go to next step.

3) If the sum of amplitudes in the current macroblock's histogram is smaller than 20000, the macroblock is homogeneous. Only check modes in the set {SKIP, 16x16, 16x8} or in the set {SKIP, 16x16, 8x16} depending on the horizontal/vertical dominant intra modes. If the dominant mode is the plane mode, only check modes in the set {SKIP, 16x16}. If the macroblock is not homogeneous, go to next step.

4) Perform ME and RDO for the 16x16, 16x8, 8x16 modes and save the best one.

5) For each 8x8 block in the macroblock, determine if the sub-block is homogeneous. If the sub-block's edge histogram is smaller than 5000, perform RDO on the 8x8 block and select 8x8 as the best mode. If not, go to next step.

6) Perform ME and RDO for the 8x8, 8x4, 4x8, 4x4 (and direct if B slice) modes and save the best mode of the sub-block.

7) Repeat steps 5 and 6 until all the best modes for the sub-blocks are found.

8) Calculate the RD cost as the sum of the four RD costs of the 8x8 sub-blocks and choose the best mode based on this sum and the RD cost of step 4.

9) Select the ultimate best mode among the best intra, the best inter and the SKIP modes. Go to step 1 and proceed with the next macroblock.

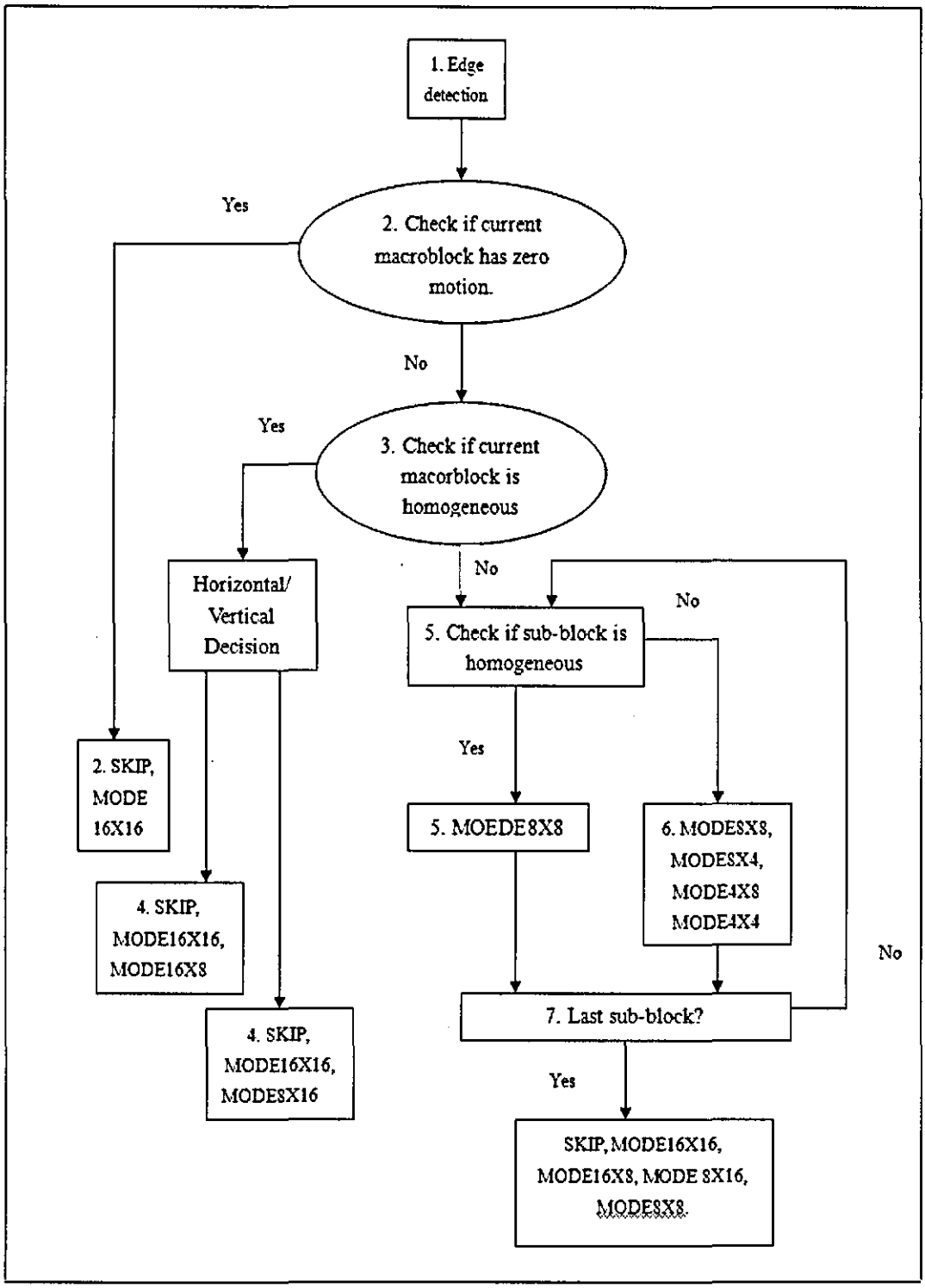


Figure 6-1 Block diagram of [67]'s algorithm

6.3 Proposed Scheme

The basic idea in the algorithm we propose can be summarised as follows: Video objects in consecutive frames are not always deformed or divided. They may be still or just change a little in position, especially for slow motion video sequences or for the slow motion frame parts of fast video sequences. In terms of computational speed ups, there is potential misprediction of modes in macroblocks of those areas from *spatial only* mode decision techniques, such as [67]. This will happen in the cases where these macroblocks have high spatial detail and as such will be assigned smaller size modes, whereas by examining error characteristics as we do, these macroblocks can be assigned larger size modes. Figure 6-2 and 6-3 provide *optimal* mode partitions of two inter frames for foreman and silent video sequences which show that not all macroblocks with high details need to check smaller size modes, for example human face parts etc. Green macroblocks are the ones which are mis-predicted by [67] and assigned to bigger size modes by H.264/AVC standard. Since in the standard, early mode assignment for larger size modes achieves computational speed-ups due to the sequential nature of the mode decision algorithm (from large to small block sizes), our error assisted mode decision algorithm can further improve the time savings of fast mode decision schemes [67] for very similar RD performance

Figure 6-2 Mode Partitions for Foreman

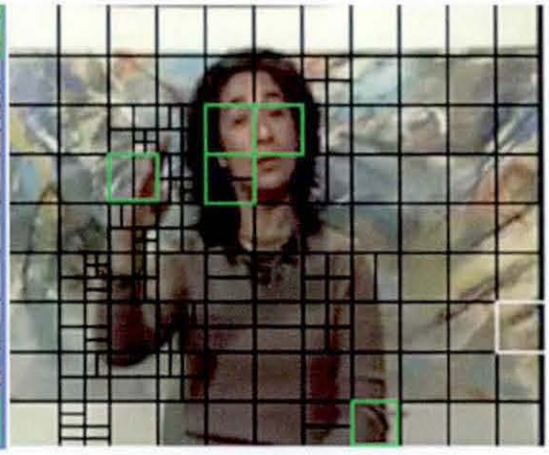


Figure 6-3 Mode Partitions for Silent

There are also two observations for the above algorithm. Firstly, the bulk of the computation is in steps 4 and 6, implying significant savings if mode assignments are made before these steps. This is achievable by considering error domain information of the best match (BM) of the current macroblock for a single ME (not exploited at all in steps 1-9) and inferences about the RDO costs of smaller block sizes. Secondly, the combination of adaptive thresholds for error homogeneity with fixed thresholds in the pixel domain [67] is expected to perform better for sequences of varying motion. The concept of the "spatial average sum of amplitudes of edge error vectors" can be exploited for designing adaptive thresholds with specific allowance for small sample sizes in the average calculation.

Considering the above observations, we insert one more step (3a) after step 3 and another step (5a) after step 5 in the above algorithm:

3a) Do ME for the current 16x16 macroblock and find its best match. Get the error macroblock and perform the same edge detection on the error domain as in [67]. Find the sum of amplitudes of the edge error vectors and if it's less than A (the average sum of amplitudes of the edge error vectors in macroblocks of mode 16x16 in the previous same slice type) deduce that the error pattern is homogeneous. Evaluate the RDO costs of {SKIP, 16x16, 16x8 and 8x16 modes}. Choose the best mode and stop. If the error pattern is non homogeneous, go to the next step.

5a) Do exactly the same as in 3a) but do ME for 8x8 block size and compare the sum of amplitudes of the edge error vectors with B, where B is the average sum of amplitudes of the edge error vectors in macroblocks of mode 8x8 in the previous slice of the same type.

The insertion of the above two steps modifies slightly steps 4 and 6 of [67], since the MEs for 16x16 and 8x8 blocks sizes do not need to be re-evaluated. In fact, the 2 MEs are only pushed in earlier steps (thus the total number of MEs remains the same) in order to make more mode assignments before the computational intensive steps are reached. Furthermore, bigger block sizes {16x16, 16x8, 8x16} are still used for mode decision in 5a) because they do not need to be split due to their homogeneous error

pattern.

Thresholds used in [67, 70] chosen in the experiments can achieve a good balance between the RD performance and time savings. We believe that [67, 70] have already obtained ideal image analysis and the remaining task is error domain analysis which will be done by the above two steps.

The above inter mode prediction scheme is illustrated with the block diagram below (figure 6-4). Two green modules are modifications for our algorithm compared with [67].

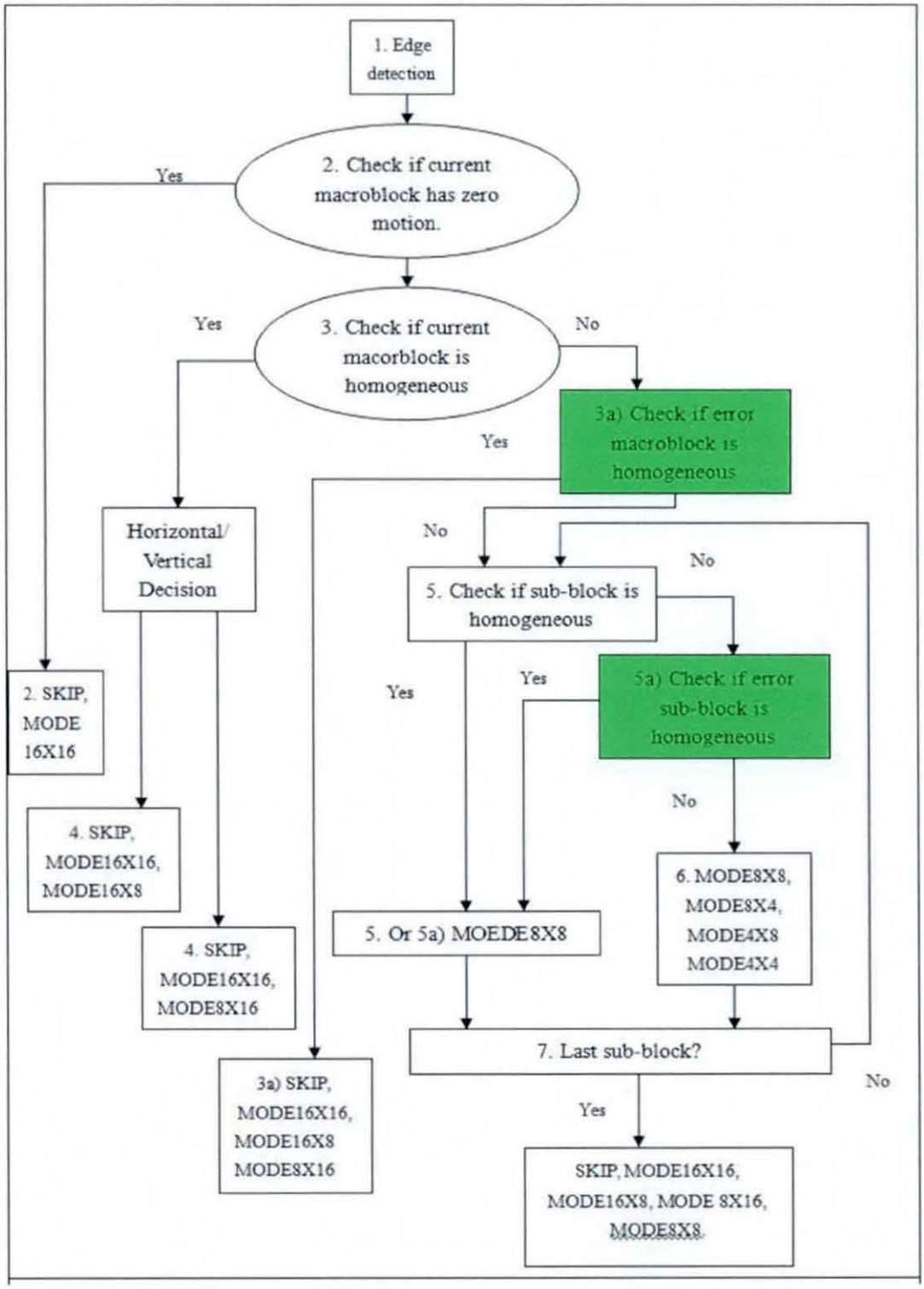


Figure 6-4 Block Diagram of Ours Algorithm

6.4 Experiments and Discussion

Experiments are performed on a Pentium 4 (3.06 GHz processor). Description of the motion characteristics of the tested video sequences has been mentioned in previous sections. One hundred frames of each sequence were examined and the encoder settings were: JM9.5, QP(28, 32, 36, 40), RD optimization ON, Hadamard Transform, IPPP and IBBP structure, no error tools, 5 reference frames and Search Range was 32. Tables 6-1 and 6-2 show the percentage time savings, the Bjontegaard Delta Bit Rate (BDBR) percentage differences and the Bjontegaard Delta PSNR (BDPSNR) differences (in dB) [96] for the baseline and main profiles between the algorithm in [67] and the enhancements we propose. Table 6-3 shows the CPU cycles needed for edge detection as a percentage of the overall number of cycles for our scheme (for different quantisation parameters, different profiles and different image resolutions). It is clear that the edge detection computation is negligible with respect to the overall computation, as expected. Graph 6-1 to 6-14 show the RD curves of original, [67]'s and our algorithm among the testing sequences. Blue line, purple line and yellow line respectively present original, [96] and our algorithm. From the graphs we can say that RD performance curves between our algorithm, [96] and original are very close. This means that comparing with original and [96] our algorithm does not degrade the coding efficiency and picture quality very much not only for the slow video sequences like News and Silent but also for the fast video sequences like Stefan and Mobile. Figure 6-5 and 6-6 show the conditional probability of correct mode prediction, given the adaptive error thresholds of steps 3a and 5a (denoted as A and B in these steps). All comparisons are with respect to the standard, QPs denote selected quantiser step sizes in the graphs and the minus signs in the tables denote PSNR

degradation and bit rate savings respectively. It can clearly be seen that comparing with [67], we can gain an average of 12% time savings for the baseline profile with BDPSNR average reduction 0.04dB and BDBR average increase 0.43%, and an average of 13% for the main profile with BDPSNR average reduction 0.03dB and BDBR average increase 0.07% depending on motion characteristics. With similar BDPSNR and BDBR we can achieve increase in time savings about 9% on the average comparing with [62] and we can gain an average increase in time savings about 19% with better Rate Distortion (RD) performance comparing with [64].

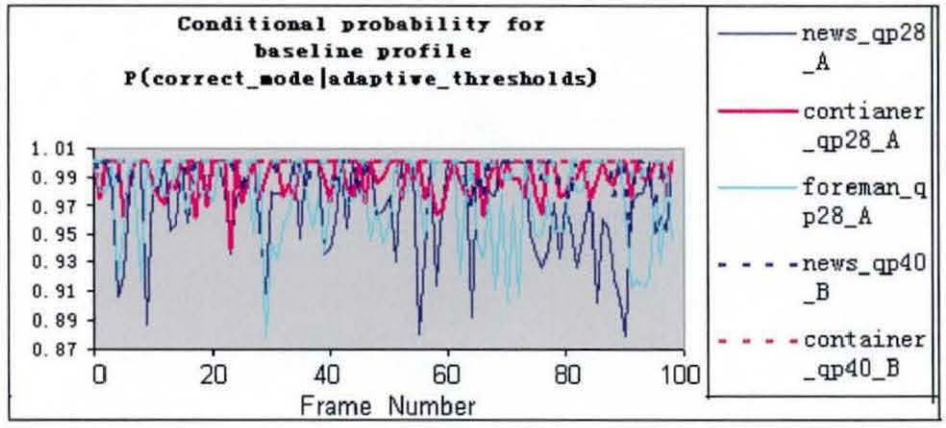


Figure 6-5

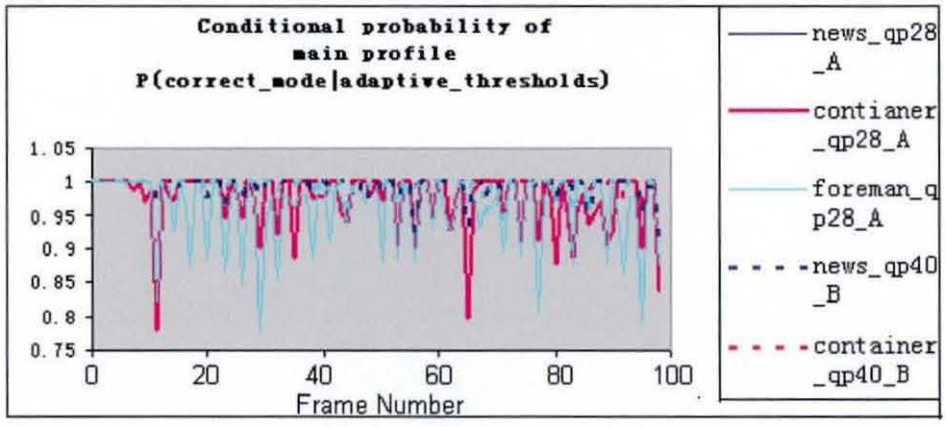


Figure 6-6

Table 6-1 Baseline profile comparison for the algorithm of Lim et. al [67] and the proposed scheme with respect to H.264/AVC

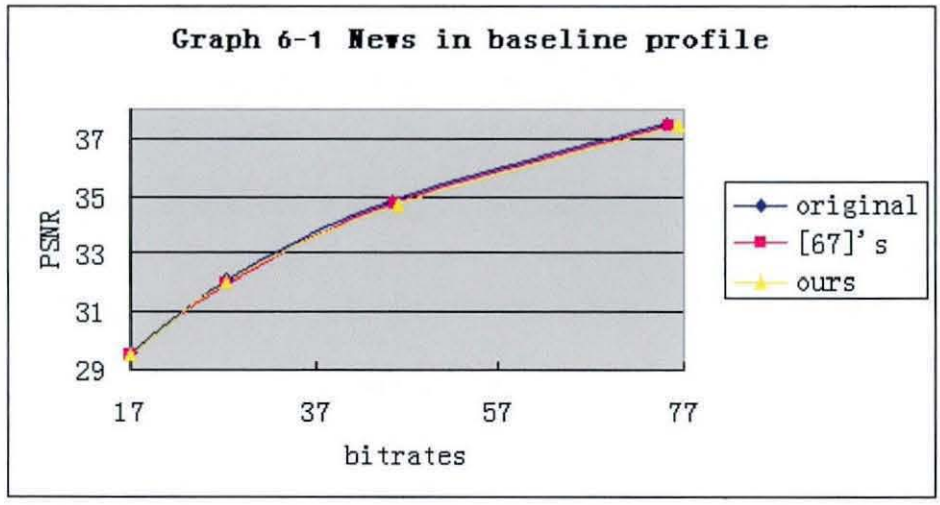
	Lim's Algorithm			*Ours algorithm*		
Sequences	***Time (%)[67]***	***BDPSNR (dB)[67]***	***BDBR (%)[67]***	***Time (%)***	***BDPSNR (dB)***	***BDBR (%)***
News(QCIF)	48.76	-0.07	0.045	54.85	-0.12	0.66
Silent(QCIF)	52.39	-0.08	0.873	56.99	-0.11	1.22
Container(QCIF)	46.04	-0.04	0.333	54.26	-0.06	0.90
Foreman(QCIF)	20.73	-0.07	1.175	36.75	-0.08	0.39
Paris(CIF)	32.83	-0.07	0.46	45.07	-0.11	0.94
Stefan(CIF)	10.02	-0.02	0.028	28.91	-0.1	1.35
Mobile(CIF)	5.64	-0.01	0	26.11	-0.05	0.47

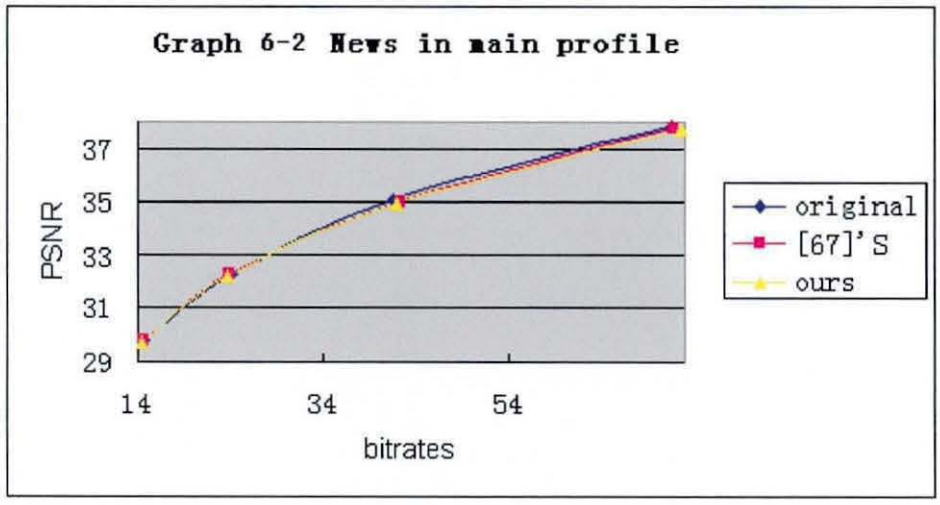
Table 6-2 Main profile comparison for the algorithm of Lim et. al [67] and the proposed scheme with respect to H.264/AVC

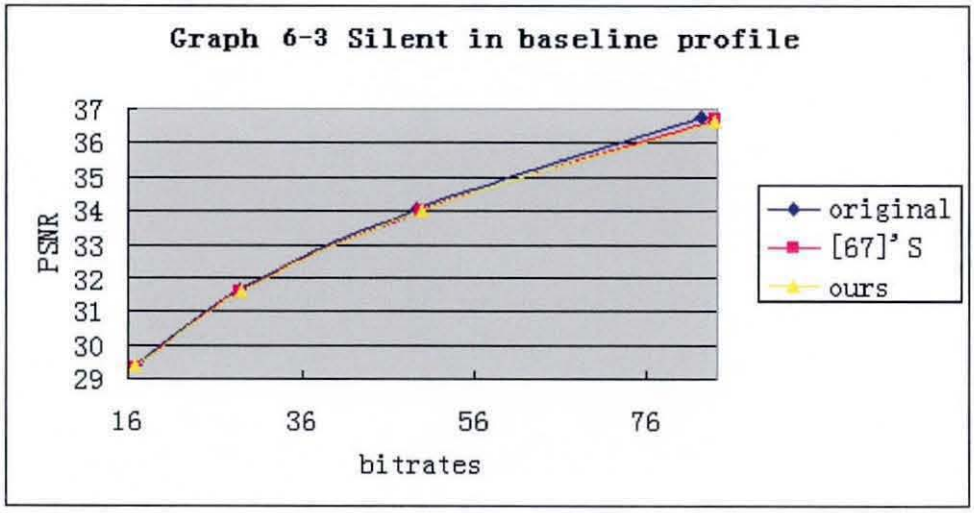
	Lim's Algorithm			*Ours algorithm*		
Sequences	***Time (%)***	***BDPSNR (dB)***	***BDBR (%)***	***Time (%)***	***BDPSNR (dB)***	***BDBR (%)***
News(QCIF)	39.48	-0.09	0.46	46.94	-0.09	-0.13
Silent(QCIF)	46.02	-0.09	0.953	52.53	-0.08	0.36
Container(QCIF)	36.74	0.01	-0.538	42.46	0.01	-0.63
Foreman(QCIF)	19.85	-0.07	-0.05	35.39	-0.1	-0.02
Paris(CIF)	23.18	-0.06	0.45	37.35	-0.1	0.76
Stefan(CIF)	9.37	-0.01	0.043	27.12	-0.15	1.46
Mobile(CIF)	5.32	-0.02	0.095	26.97	-0.07	0.1

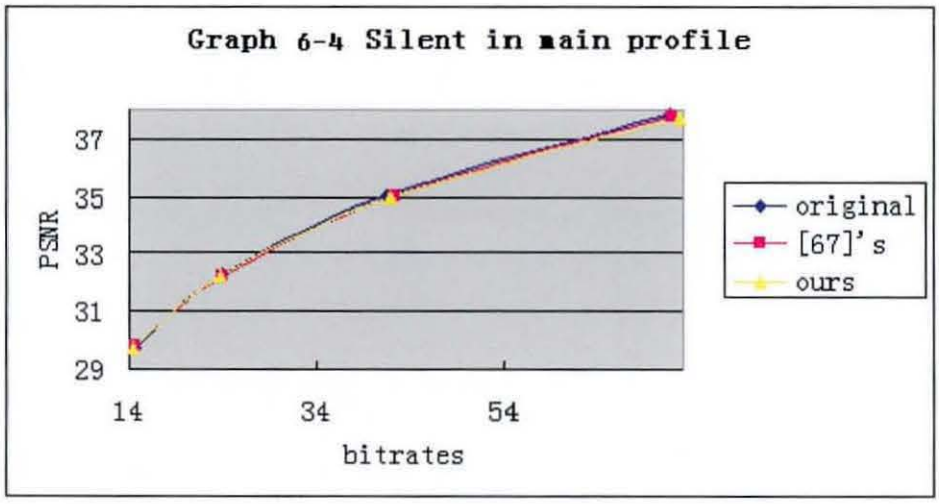
Table 6-3: CPU cycles spent on the edge detection part of our scheme as a percentage of the total number of cycles using different quantisation parameters(measured with the Vtune tool for Pentium Processors)

	Cycles(%)			
Sequence	QP 28	QP 32	QP 36	QP 40
News(QCIF/Simple)	0.24	0.17	0.15	0.16
Silent(QCIF/Simple)	0.18	0.15	0.21	0.18
Container(QCIF/Simple)	0.18	0.16	0.16	0.15
Foreman(QCIF/Simple)	0.07	0.13	0.08	0.14
News(QCIF/Main)	0.13	0.12	0.13	0.09
Silent(QCIF/Main)	0.08	0.10	0.12	0.13
Container(QCIF/Main)	0.12	0.13	0.16	0.20
Foreman(QCIF/Main)	0.06	0.04	0.17	0.09

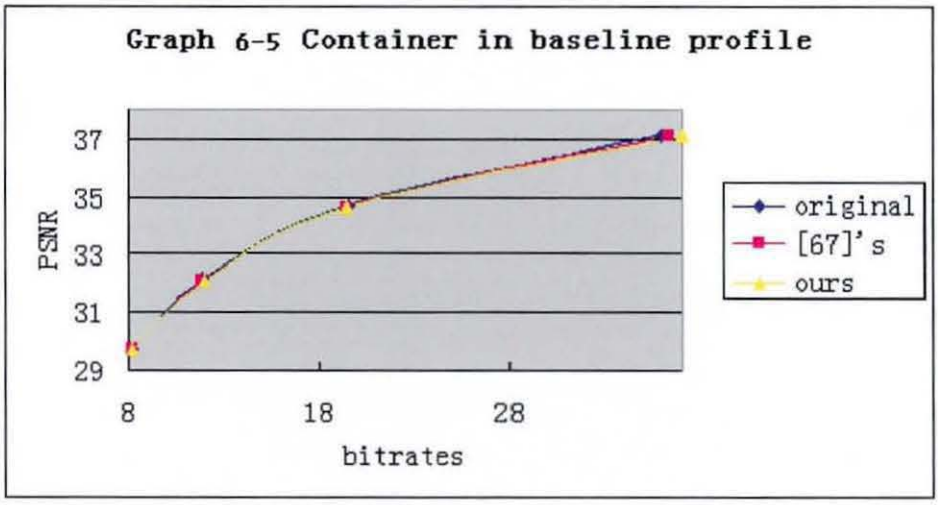

Graph 6-1 News in baseline profile
PSNR
37
35
33
31
29
17
37
57
77
bitrates
original
[67]'s
ours


Graph 6-2 News in main profile
PSNR
37
35
33
31
29
14
34
54
bitrates
original
[67]'S
ours

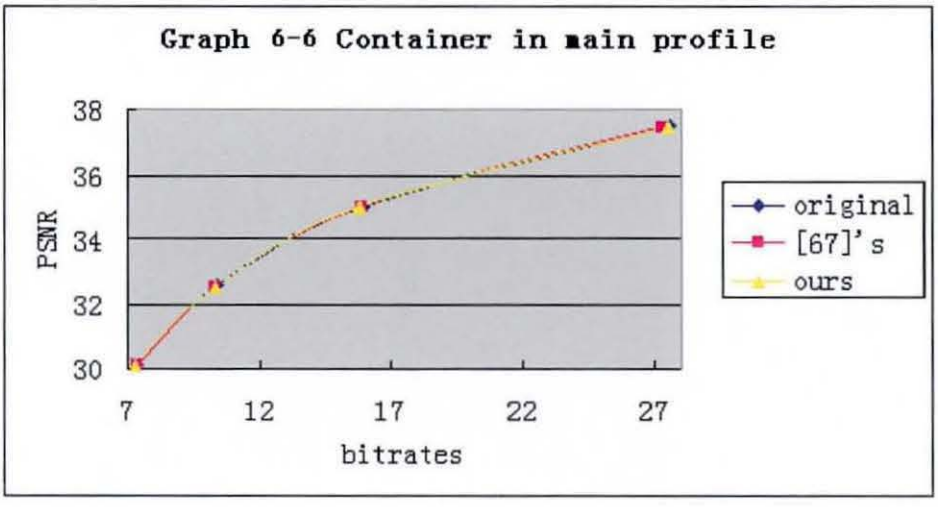

Graph 6-3 Silent in baseline profile
PSNR
37
36
35
34
33
32
31
30
29
16
36
56
76
bitrates
original
[67]'S
ours



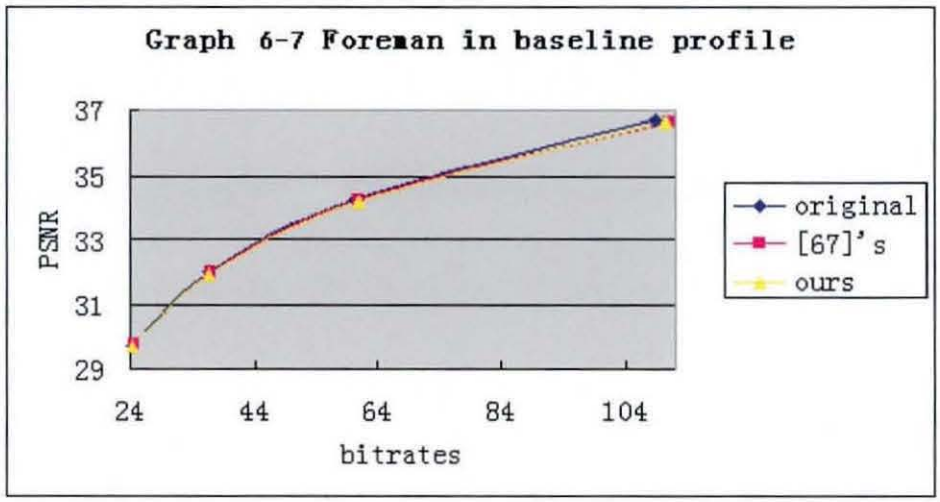
Graph 6-4 Silent in main profile

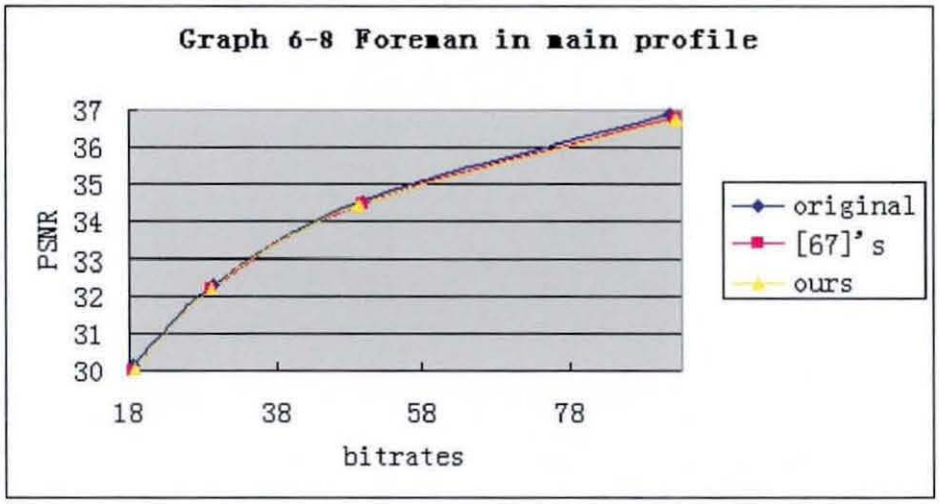


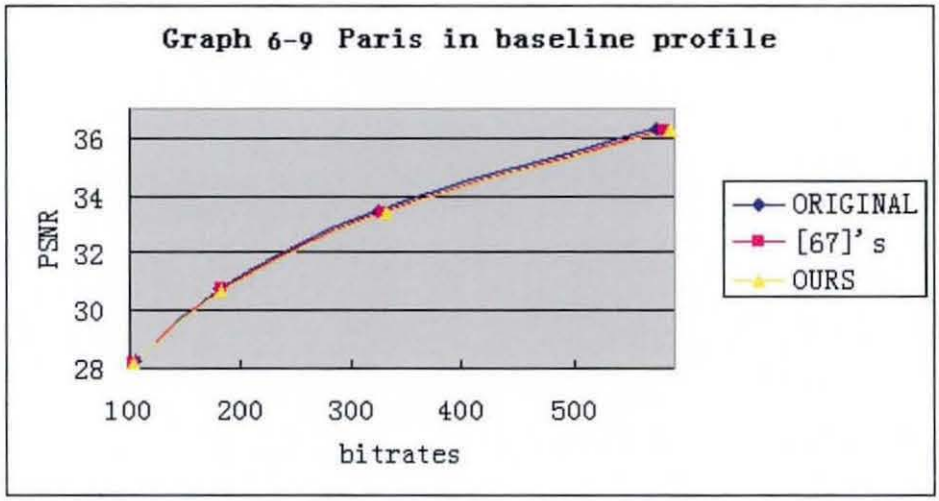
Graph 6-5 Container in baseline profile

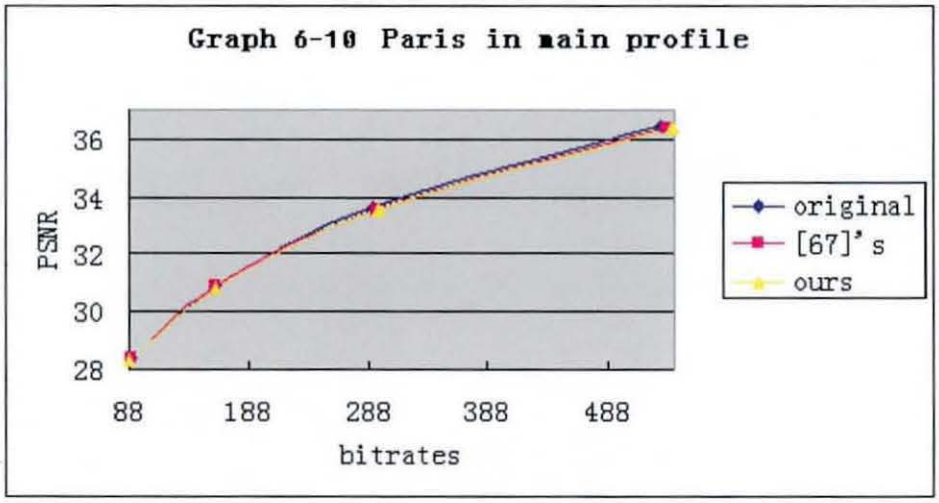


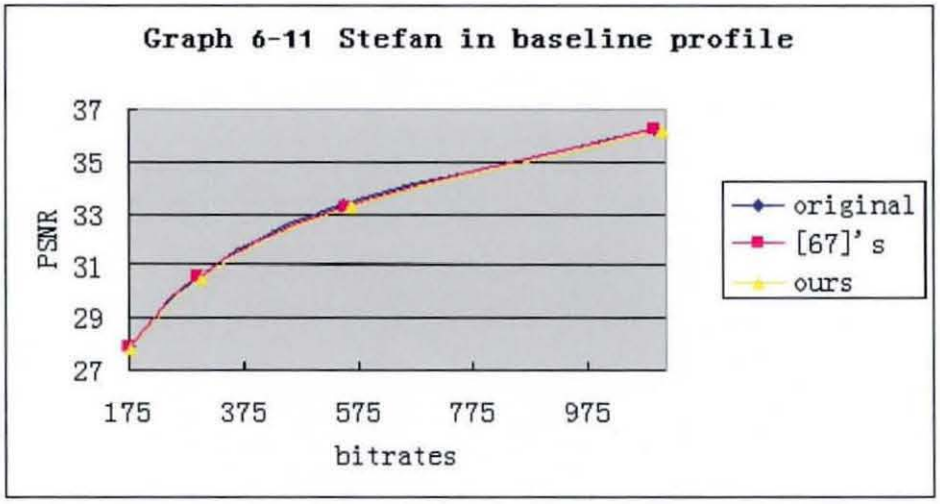
Graph 6-6 Container in main profile

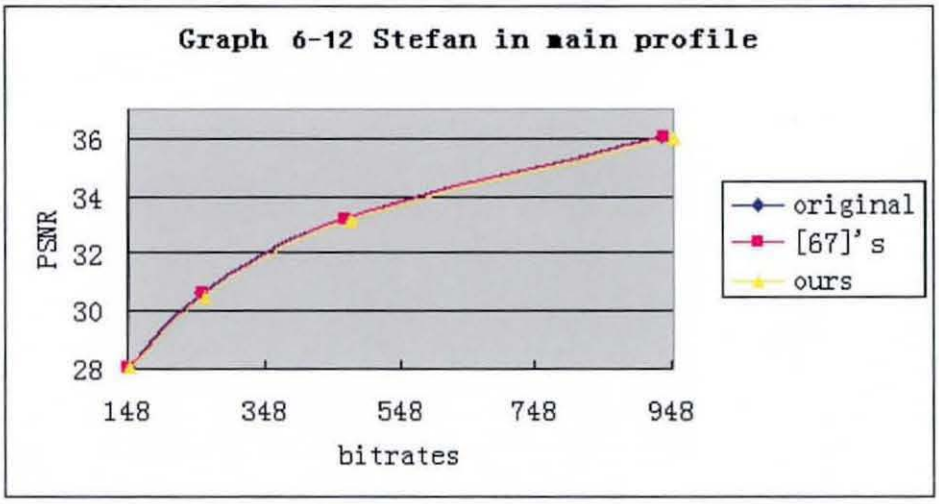

Graph 6-7 Foreman in baseline profile
PSNR
37
35
33
31
29
24
44
64
84
104
bitrates
original
[67]'s
ours

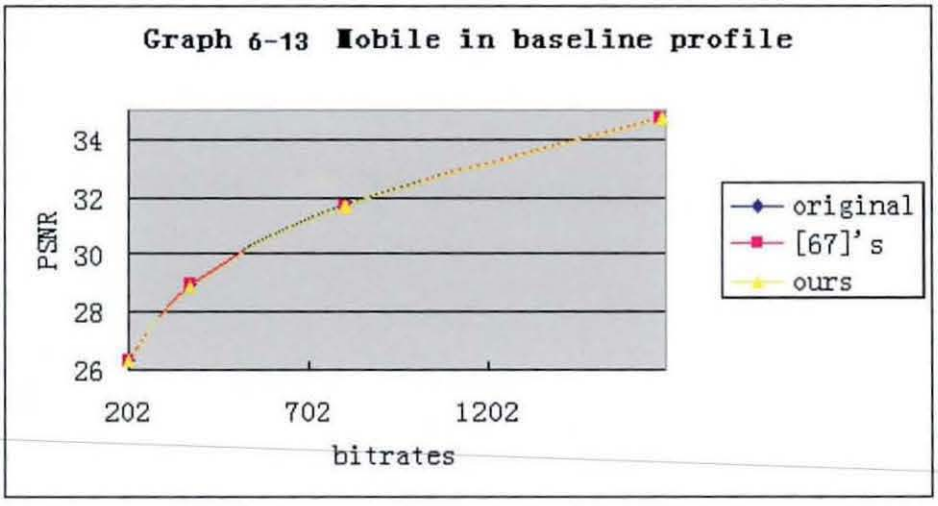

Graph 6-8 Foreman in main profile
PSNR
37
36
35
34
33
32
31
30
18
38
58
78
bitrates
original
[67]'s
ours

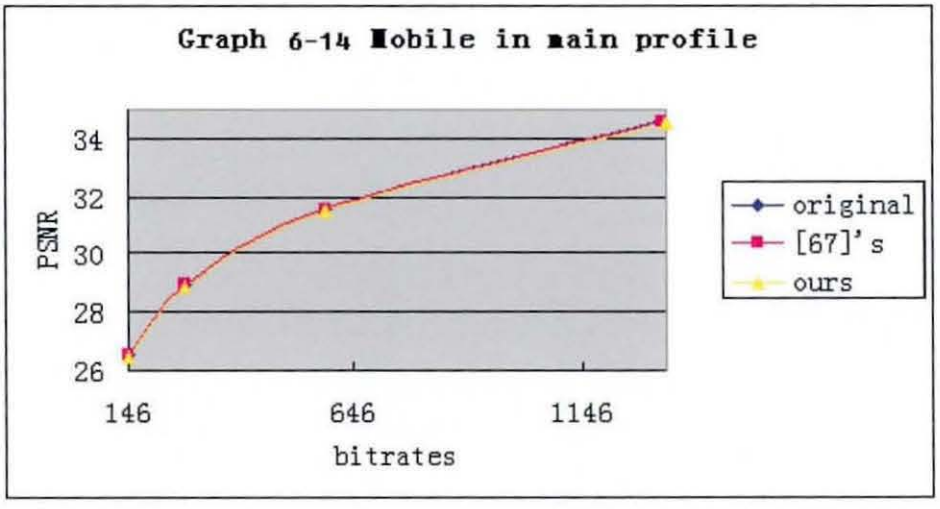

Graph 6-9 Paris in baseline profile
PSNR
36
34
32
30
28
100
200
300
400
500
bitrates
ORIGINAL
[67]'s
OURS


Graph 6-10 Paris in main profile
PSNR
36
34
32
30
28
88
188
288
388
488
bitrates
original
[67]'s
ours


Graph 6-11 Stefan in baseline profile
PSNR
37
35
33
31
29
27
175
375
575
775
975
bitrates
original
[67]'s
ours


Graph 6-12 Stefan in main profile
PSNR
36
34
32
30
28
148
348
548
748
948
bitrates
original
[67]'s
ours


Graph 6-13 Mobile in baseline profile
PSNR
34
32
30
28
26
202
702
1202
bitrates
original
[67]'s
ours


Graph 6-14 Mobile in main profile
PSNR
34
32
30
28
26
146
646
1146
bitrates
original
[67]'s
ours

6.5 Conclusion

To summarise, in this chapter, we have proposed a fast mode decision scheme for the Baseline and Main profiles of the H.264/AVC standard. Our fast mode decision scheme is improvement of [67] which is adopted by H.264/AVC standard. In [67] only spatial analysis information has been used to predict the modes, but in our algorithm we not only use spatial information but also use the temporal information which is exploited in the error domain to predict modes. Graphs are provided to prove that RD performance of our algorithm is very close to the standard and tables are provided to show that comparing with H.264/AVC standard our scheme can achieve an average of 43% time savings in baseline profile and 38% time savings in main profile. Our scheme can be a very useful technique for low complexity video coding systems like video conference and mobile.

Chapter 7

Thesis Conclusion

7.1 Preamble

Chapter 6 is the final chapter of this thesis. This chapter draws conclusions and summarises our contributions to the field of fast mode prediction in the H.264/AVC standard. Limitations of the proposed schemes and directions for future research are also presented in the end.

7.2 Contributions

In this thesis we contributed three fast mode decision techniques which can be classified into two categories, frameworks of fast mode decision and error domain fast mode decision.

In chapter 4, two fast mode decision algorithms have been proposed, fast inter mode prediction for P slice [91] and fast mode prediction for baseline and main profiles [97]. In the former algorithm, initially the property of skip mode clustering is exploited to predict the skip mode without doing any motion estimation. JEON's skip mode detection [55] is preformed in the second step. Finally a gentle set of smoothness constraints is performed to predict inter modes for the remaining macroblocks. Experiments show that 35%-58% of encoding time saving can be achieved with very similar rate distortion performance compared with H.264/AVC standard. The central observation of this algorithm is that temporal RD cost correlation is a very good indicator for us to predict modes of macroblocks with big partition block sizes. A lot of previous contributions for fast mode decision use simply MAD or SAD information, motion vectors, or image feature analysis

and so on which either lose some video quality or can not save too much time. There are two advantages to use the RD cost indicator. One is that this indicator is significantly correlated temporally and as such more trustworthy. Another is that this indicator is adaptive and depends on video contents, as such it can predict many more modes accurately. The second algorithm can be considered as an extension of the former one for both the baseline and main profiles of the H.264/AVC standard. This scheme utilises a set of skip mode conditions for P and B slices [55, 56], two heuristics that reduce the cardinality of the inter mode set to be examined, inter/intra mode prediction and the monotonicity property [77] of the Rate Distortion cost functions. Using this algorithm 33-90% of content dependent time saving can be obtained with very similar rate distortion performance.

The error domain fast mode decision algorithm [98] which is mentioned in chapter 5 is based on the algorithm of [67]. In the proposed scheme, the edge detection technique is not only implemented on the image domain but also on the error domain. The proposed improvement is based on the fact that in the image domain, edge detection can only detect smooth parts of an image and also needs a threshold. If it is applied in the error domain too though, more macroblocks can be predicted, especially with the use of an adaptive threshold based on video contents. Experiments demonstrate that at the same bit rate, this algorithm can achieve bigger time savings compared with [67], with negligible video quality loss.

7.3 Limitations and Future Work

We have to admit there are still limitations in our schemes. For example, Graphs 4-17 to 4-26 demonstrate that our fast mode decision scheme in the main profile, rate controlled case cannot achieve as good RD performance as in other cases. One of the problems is the skip mode decision technique [56]. Table 5-10 shows that [56] in main profile does not work so well compared with [55] in baseline profile, especially for fast motion video sequences. Because B slices contain even more skip mode macroblocks than P slices, an improved skip mode detection in B slices is worth to be developed based on special characteristics of B slices.

Another problem is that none of our and previous fast mode decision schemes consider about quantization parameters changing in the rate control case. I did some experiments about the average RD cost or moving average sum of amplitudes of edge error vectors of the macroblocks in the same mode in different parts of the reference frame. I found that the average values in different parts were different. For example, in Figure 7-1, the average value of the left part of this frame is smaller than the average value of the right part. That implies that if we use the global average value to judge smoothness in the left part, misjudgments will occur in the right part since both parts are not equally smooth. This will affect not only the PSNR performance and the time-savings but also the bit rates. Using local information to substitute global information is a good idea to overcome this problem. But how to set the local window size will become a problem.

Figure 7-1 Carphone.qcif

Rate distortion performance can be improved in H.264/AVC video coding standard by using multiple reference frames. This new feature may be also helpful for fast mode prediction techniques which I have proposed. Average Rate distortion costs and amplitudes of edge error vectors from the first previous frame are used to be adaptive thresholds to predict modes in our algorithms. If we can extend to calculate the thresholds from several instead of just one previous reference frames and use different weights

between these frames based on the distances from current frame, hopefully we can get even better fast mode prediction techniques.

Scalable video coding (SVC) has been a hot research topic for about 20 years and it enables an encoder to arrange the coded stream in a number of layers, including a base layer and several enhancement layers. In January 2005, JVT planned to finalize the SVC project as an Amendment of their H.264/AVC standard, and the scalable coding scheme developed by Heinrich Hertz Institute (HHI) of Germany was adopted as the beginning working draft (WD). In the WD, exhaustive search technique is employed to select the best coding mode for each macorblock in base layer and enhancement layers. This mode decision technique achieves higher coding efficiency but results in higher computational complexity. Apparently, mode relationship between base layer and enhancement layers contains high correlation which can be used to reduce the candidate mode set at enhancement layers. Future research work about fast mode decision can also focus on this direction.

References

[1] J.R. Jain and A.K. Jain, "*Displacement measurement and its application in interframe image coding*", IEEE Trans. Commun., Vol. COM-29, No. 12, pp. 1799-1808, Dec. 1981.

[2] J. Ribas-Corbera and D.L. Neuhoff "*On the optimal block size for block-based, motion compensated video coders*", SPIE Proceedings of Visual Communications and Image Processing, Vol. 3024, pp 1132-1143, February 1997.

[3] M.H. Chan, Y.B. Yu, A.G. Constantinides, "*Variable size block matching motion compensation with applications to video coding*", IEE Proceedings, Vol 137, Pt. 1, No. 4, August 1990.

[4] T Koga, K Iinuma, A Hirano, Y Iijima, and T Ishiguro, "*Motion compensated interframe coding for video conferencing*," *Proc. National Telecommunication Conference*, pp. G.5.3.1-5.3.5, 29 Nov. - 3 Dec. 1981.

[5] Lai-Man Po, Wing-Chung Ma "*A Novel Four Step Search Algorithm For Fast Block Motion Estimation*" IEEE Transactions on Circuits and Systems for Video Technology, vol. 6, no.3, pp 313-7, June 1996.

[6] M.Ghanbari "*The cross search algorithm for Motion Estimation*" IEEE Transactions on Communications, vol. 38, no. 7 pp 950-3, July 1990.

[7] Th. Zahariadis, D. Kalivas "*A Spiral Search Algorithm For Fast Estimation Of Block Motion Vectors*" Signal Processing VIII, theories and applications. Proceedings of the EUSIPCO 96. Eighth European Signal Processing Conference p.3 vol. lxiii + 2144, 1079-82, vol. 2.

[8] Kwon Moon Nam, Joon-Seek Kim, Rae-Hong Park "*A Fast Hierarchical Motion Vector Estimation Algorithm Using Mean Pyramid*" IEEE Transactions on Circuits and Systems for Video technology, vol. 5, no.4, pp 344-351, August 1995.

[9] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "*Motion compensated interframe coding for video conferencing*," in *Proc. Nat. Telecomm. Conf.*, pp. G5.3.1–G5.3.5, New Orleans, LA ~1981.

[10] S Kappagantula and K R Rao, "*Motion compensated predictive coding*," *Proc. SPIE 27th*, pp. G4-70, 1983.

[11] Anders Vretblad, "*Fourier Analysis and Its Applications*," Springer-Verlag, New York, 2000.

[12] D Sundararajan, "*The Discrete Fourier Transform: Theory, Algorithms and Applications*," World Scientific, 2001.

[13] K. R. Rao and P. C. Yip, "*The Transform and Data Compression Handbook*," CRC Press LLC, USA, 2000.

[14] L Prasad, S S Ayengar, S S Iyengar, "*Wavelet Analysis With Application to Image Processing*," CRC Press LLC, USA, 1997.

[15] K. R. Rao and P. C. Yip, "*Discrete Cosine Transform: Algorithms, Advantages, Applications*," Academic Press, London, 1990.

[16] David Salomon, "*Data Compression: The Complete Reference*," Springer-Verlag, New York, 1998.

[17] Allen Gersho, Robert M. Gray, "*Vector Quantization and Signal Compression*," Kluwer Academic, USA, 1992.

[18] John Watkinson, "*The Mpeg Handbook*," Focal press, Elsevier Science, USA, 2001

[19] DA Huffman, "*A method for the construction of minimum redundancy codes*," Proc. IRE, no. 40, pp. 1098–1101, Sept. 1952.

[20] Rissanen, J., Langdon, G. G, "*Arithmetic coding,*" IBM Journal of Research and Development, 23, 149-162. 1979.

[21] I. Witten, R. Neal, and J. Cleary. Arithmetic coding for data compression. Comm. of the ACM, 30(6), June 1987.

[22] Golomb SW, "*Run-length encodings*," IEEE Transactions on Information Theory, 12(3):399—401. 1966.

[23] ITU-R Recommendation BT.500-10. "*Methodology for the subjective assessment of the quality of television pictures*," March 2000.

[24] A. M. Eskicioglu and P. S. Fisher, "*Image quality measures and their performance*,"*IEEE Transactions on Communications*, vol. 43, no. 12, pp. 2959-2965, Dec. 1995.

[25]A. M. Mayache, T. Eude, and H. Cherifi, "*A comparison of image quality models and metrics based on human visual sensitivity*," IEE E International Conference on *Image Processing*, vol. 3, pp. 409-413, Oct. 1998.

[26]B. Girod, "Psychovisual aspects of image communication," Signal Processing, vol. 28, no. 3, pp. 239-251, Sept. 1992.

[27]T. Wiegand, G.J. Sullivan, G. Bjntegaard, A. Luthra "*Overview of the H.264/AVC video coding standard*," IEEE Transactions on Circuits and Systems for Video Technology, vol 13, no 7, July 2003.

[28]S. Wenger, "*H.264/AVC over IP*," IEEE Transactions on Circuits and Systems for Video Technology, vol 13, no 7, July 2003.

[29]"*Generic Coding of Moving Pictures and associated Audio Information. Part 2: Video*", ITU-T and ISO/IEC JTC1, ITU-T Recommendation ISO/IEC 13818-2 (MPEG-2) 1994.

[30]"*Coding of Audio Visual Objects Part 2: Visual*", ISO/IEC JTC1, ISO/IEC 14496-2 (MPEG-4 Visual version 1), 1999.

[31]"*Video Coding for Low Bit Rate Communication*," ITU-T Recommendation H263, version 2, January 1998.

[32]Jiri Jan, "*Digital Signal Filtering, Analysis and Restoration*," the institution of electrical engineers, London, UK, 2000.

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

[34]G. J. Sullivan, P. Topiwala and A. Luthra "*The H.264/AVC advanced video coding standard: Overview and introduction to the fidelity range extensions*", SPIE Conf. on applications of digital image processing XXVII, vol. 5558, pp. 53-74, Aug. 2004.

[35]D. Marpe, V. George, H.L. Cycon, and K.U. Barthel, "*Performance evaluation of Motion-JPEG2000 in comparison with H.264/AVC operated in intra coding mode*," in Proc. SPIE Conf. on Wavelet Applications in Industrial Processing, Photonics East, Rhode Island, USA, Oct. 2003.

[36]ISO/IEC 15444-3 Motion-JPEG2000 (JPEG2000 Part 3), 2002.

[37]M.Wien, "Variable Block-Size Transforms for H.264/AVC," IEEE Trans.Circuits and System. Video Technology, vol. 13, pp. 604-613, July 2003.

[38]M.Wien, "*Adaptive Deblocking Filter*", IEEE Trans.Circuits and System. Video Technology, vol. 13, pp. 614-619, July 2003.

[39]D.Marpe, H.Schwarz, T. Wiegand, "*Context-Based Adaptive Binary Arithmetic Coding in the H.264/AVC Video Compression Standard*," IEEE Trans.Circuits and System. Video Technology, vol. 13, pp. 620-636, July 2003.

[40]S. Saponara, K. Denolf, C. Blanch, G. Lafruit, J. Bormans, "*Performance and complexity co-evaluation of the Advanced Video Coding standard for cost-effective multimedia communications*", EURASIP Journal of Applied Signal Processing, 2003.

[41]A Joch, F Kossentini, P Nasiopoulos, "*A performance analysis of the ITU-T draft H. 26L video coding standard*", 12th International Packetvideo Workshop, 2002.

[42]J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockammer and T. Wedi, "*Video Coding with H.264/AVC: Tools, Performance, and Complexity*", IEEE Circuits and Systems, Vol. 4, No. 1, 2004.

[43]http://iphome.hhi.de/suehring/tml/download/old_jm/ jm81a.zip

[44]C.E. Shannon, "*A mathematical theory of communication*," Bell Syst. Tech. Journal, vol. 27, pp. 379- 423, 1948.

[45]C.E. Shannon, "*Coding theorems for a discrete source with a fidelity criterion*," in IRE National Convention Record, Part4, pp. 142-163, 1959.

[46]T. Berger, "*Rate-Distortion Theory. A Mathematical Theory Data Compression,*" Prentice-Hall, 1971.

[47]G. M. Schuster and A. K. Katsaggelos, "*Rate-Distortion Based Video Compression, Optimal Video Frame Compression and Object Boundary Encoding,*" Boston, MA: Kluwer, 1997.

[48]A. Ortego and K. Ramchandran, "*Rate-distortion methods for image and video compression,*" IEEE Signal Processing Magazine, vol. 15, pp. 23--50, 1998.

[49]G. J. Sullivan and T. Wiegand, "*Rate-Distortion Optimization for Video Compression*", IEEE Signal Processing Magazine, vol. 15, no. 6, pp. 74--90, Nov. 1998.

[50]Bertsekas, D. P., "*Constrained Optimization and Lagrange Multiplier Methods*", Academic Press, 1982

[51]Arfken, G. "Lagrange Multipliers." in Mathematical Methods for Physicists, 3rd ed. Orlando, FL: Academic Press, pp. 945-950, 1985.

[52]R. Fletcher. "Practical methods of optimization", Wiley, Chichester; New York, second edition, 1987

[53]Y. Zhao and I. E. G. Richardson, "*Macroblock classification for complexity management of video encoders*," *Signal Process.: Image Commun.*, vol. 18, pp. 801–811, Oct. 2003.

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

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

[56]J.Lee, I.Choi, W.Choi and B.Jeon, "*Fast Mode Decision for B slice*", ISO/IEC JTC1/SC29/WG11 and ITU-T SG16, Input Document JVT-K021, March 2004.

[57]I.Choi, W.Choi, J.Lee and B. Jeon , "*The Fast Mode decision with Fast Motion Estimation*", ISO/IEC JTC1/SC29/WG11 and ITU-T SG16, Input Document JVT-N013, Jan 2005.

[58]Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, "*Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC)*," Doc. G050r1, Mar. 2003.

[59]M. Chan Y. Yu, and A. Constantinides, "*Variable sizeblock matching motion compensation with applications to video coding*," *Proc. Inst. Elect. Eng.*, pt I, vol. 137, no. 4, pp. 205-212, Aug. 1990.

[60]I. Rhee, G. Martin, S. Muthukrishnan, and R. Packwood, "*Quadtree-Structured Variable-Size Block-Matching Motion Estimation with Minimal Error*," IEEE Trans. Circuirs Syst. Video Technol., vol.10, pp.42–50, Feb. 2000.

[61]Yu-Kuang Tu, Jar-Ferr Yang Yi-Nung Shen, and Ming-Ting Sun, "*Fast variable-size block motion estimation using merging procedure with an adaptive threshold*," IEEE ICME 2003, pp789-792

[62] X. Jing, L.P. Chau, "*Fast Approach for H.264 Inter Mode Decision*," Electronics letters, Vol. 40, Issue 17, Aug 2004.

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

[64] A. Ahmad, N. Khan, S. Masud and M.A. Maud, "*Efficient Block Size Selection in H.264 Video Coding Standard*," Electronics letters, vol 40, issue 1, Jan 2004

[65] De Haan, G., Biezen, P.W.A.C., Ojo, O.A., and Huijgen, H.: 'True motion estimation with 3-D recursive search block matching', IEEE Trans. Circuits Syst. Video Technol., 1993, 3, pp. 368–379, 388

[66] A.C Yu, "*Efficient Block-size Selection Algorithm for Inter-frame Coding in H.264/MPEG-4 AVC*," IEEE International Conference on Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04), vol 3, 17-21 May 2004.

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

[68] B. K. P Horn and B. G. Schunk. "*Determining optical flow*," Artificial Intelligence, 17:185—203 1981

[69] A. M. Tekalp, "*Digital Video Processing*," Prentice Hall, 1995

[70] F.Pan, X.Lin, R.Susanto, K.P.Lim, Z.G.Li, G.N.Feng, D.J.Wu and S.Wu, "*Fast Mode Decision for Intra Prediction*", ISO/IEC JTC1/SC29/WG11 and ITU-T SG16, Input Document JVT-G013

[71] D. Zhu, Q. Dai, R. Ding, "*Fast Inter Prediction Mode Decision for H.264*," IEEE International Conference on Multimedia and Expo, 2004. ICME '04. vol 2, 27-30 June 2004.

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

[73] Ludmila I. Kuncheva, "*Ruzzy classifier design,*" chapter5, Physica-Verlag Heidelberg, New York, 2000.

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

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

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

[77]P.Yin, H.Y.Cheong, A.Tourapis and J.Boyce, "*Fast Mode Decision and Motion Estimation for JVT/H.264*",IEEE International Conference in Image Processing 2003

[78]B. Meng, O.C. Au, C. Wong, and H. Lam, "*Efficient intra-prediction mode selection for 4x4 blocks in H.264*", 2003 IEEE lot. Conf. Multimedia &Expo (ICME2003). Baltimore. Maryland, USA, July *2003*

[79] Y. Zhang, F. Dai, S. Lin, "*Fast 4x4 Intra-prediction Mode Selection for H.264*," 2004 IEEE International Conference on Multimedia and Expo (ICME2004), Vol 2, 27-30, June 2004.

[80]Rafael C. Gonzalez, Richard E. Woods, "*Digital Image Processing*," Prentice Hall, 2002

[81]F. Fu, X, Lin, L. Xu, "*Fast Intra Prediction Algorithm in H.264/AVC*," 2004 7th International Conference On Signal Processing, 2004, Proceedings, ICSP'04, Vol 2, 31Aug-4Sept, 2004.

[82]C. Yang, L. PO, W. Lam, "*A Fast H.264 Intra Prediction Algorithm using Macroblock Properties*," 2004 International Conference on Image Processing, ICIP'04, vol 1, 24-27 Oct. 2004.

[83]Changsung Kim, Hsuan-Huei Shih and C. C. Jay Kuo, "*Multistage Mode Decision for Intra Prediction in H.264 Codec*," IS&T/SPIE 16th Annual Symposium EI, Visual Communications and Image Processing, Orlando, Florida, January, 2004.

[84]Changsung Kim, Hsuan-Huei Shih and C. C. Jay Kuo, "*Feature-Based Intra-Prediction Mode Decision for H.264*", IEEE Proceedings of International Conference Image Processing, *submitted*, Singapole, October, 2004.

[85]Changsung Kim, Hsuan-Huei Shih, C.-C. Jay Kuo, "*Fast H.264 Intra-Prediction Mode Selection Using Joint Spatial and Transform Domain Features*," Journal of Visual Communication and Image Representation, April, ELSEVIER, 2006.

[86]C. Tseng, H. Wang, J. Yang, "*Improved and Fast Algorithms for Intra 4x4 Mode Decision in H.264/AVC*," IEEE International Symposium on Circuits and System, 2005, ISCAS 2005. May, 2005.

[87]C. Cheng, T. Chang, "*Fast Three Step Intra Prediction Algorithm for 4x4 Blocks in H.264*," IEEE International Symposium on Circuits and System, 2005, ISCAS 2005. May, 2005.

[88]Y.-K. Chen, A. Vetro, H. Sun, and S. Y. Kung, "*Optimizing intra/inter coding mode decisions*," in Proc. 1997 International Symposium on Multimedia Information Processing, Taipei, Taiwan, 1997

[89]D. S. Turaga and T. Chen, "*Classification based mode decisions for video over networks*," *IEEE Trans. Multimedia*, vol. 3, no. 1, pp. 41–52, 2001.

[90]Changsung Kim and C. C. Jay Kuo, "*A Feature-based Approach to Fast H.264 Intra/Inter Mode Decision*", IEEE Proceedings of International Symposium on Circuit and System, Kobe in Japan, May 2005.

[91]Christos Grecos, Mingyuan Yang, "*Fast inter mode prediction for P slices in the H264 video coding standard*," IEEE Transaction on Broadcasting, Volume 51, Issue 2, June 2005

[92]T.Uchiyama, N. Mukawa and H.Kaneko, "*Estimation of Homogeneous Regions for Segmentation of Textures Images*", IEEE Proceedings in Pattern Recognition and Machine Intelligence, 2000, pp. 1072-1075.

[93]K. R. Castleman, "*Digital Image Processing*," Prentice Hall, 1996.

[94]A.M.Bazen and S.H.Gerez, "*Systematic Methods for the Computation of the Directional Fields and Singular Points in Fingerprints*", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, pp. 905-919, July 2002.

[95]J. Lee and B. Jeon, "*Fast Mode Decision for H264 with Variable Motion Block Sizes*", ISCIS 2003, Lecture Notes in Computer Science 2869, pp. 723-730, Springer-Verlag, 2003.

[96]G. Bjontegaard, "*Calculation of Average PSNR Differences between RD-curves*," Doc. VCEG-M33 , Apr. 2001.

[97]Christos Grecos, Mingyuan Yang, *"Fast mode prediction for the Baseline and Main profiles in the H.264 video coding standard,"* accepted for publication in IEEE Transaction on Multimedia, 2006

[98]Mingyuan Yang, Chirstos Grecos, Lihui Chen, *"Reduced Computation Mode decision using Error Domain heuristics for the H264 standard,"* IEEE Ph.D. Research in Microelectronics and Electronics, Otranto (Lecce) in Italy, June 2006.

[99]Ce Zhu, Xiao Lin, and Lap-Pui Chau, "Hexagon-Based Search Patten for Fast Block Motion Estimation", IEEE Trans. on CSVT, pp.349-355, Vol.12, No.5, May, 2002.

[100]Zhibo Chen, Peng Zhou, Yun He, "Fast Integer Pel and Fractional Pel Motion estimation in for JVT", JVT-F017r1.doc, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, 6th meeting, Awaji, Island, JP, 5-13 December, 2002

[101]Zhibo Chen, etc. "Hybrid Unsymmetrical-cross Multi-Hexagon-grid Search Strategy for Integer Pel Motion Estimation in H.264", Picture Coding Symposium, 2003, April, Saint_Malo, France

[102]http://www.cs.tut.fi/~karaoglu/lectures/DIP3_H264.pdf

Selected Publications

Journals:

Christos Grecos, Mingyuan Yang, "*Fast inter mode prediction for P slices in the H264 video coding standard,*" IEEE Transaction on Broadcasting, Volume 51, Issue 2, June 2005.

Christos Grecos, Mingyuan Yang, "*A Framework for Fast Mode Decision in the H264 video coding standard,*" accepted for publication in Digital Signal Processing, elsevier, 2006

Christos Grecos, Mingyuan Yang, "*Fast mode prediction for the Baseline and Main profiles in the H.264 video coding standard,*" accepted for publication in IEEE Transaction on Multimedia, 2006.

Christos Grecos, Mingyuan Yang, "*Exploiting spatio-temporal information for fast mode decision in the H264 video coding standard,*" accepted for publication in Multidimensional Systems and Signal Processing, Springer, 2006

Conferences:

Mingyuan Yang, Chirstos Grecos, Lihui Chen, "*Reduced Computation Mode decision using Error Domain heuristics for the H264 standard,*" IEEE Ph.D. Research in Microelectronics and Electronics, Otranto (Lecce) in Italy, June 2006.