THE UNIVERSITY OF
WARWICK

Annual Report

EFFICIENT INTRA- AND INTER-MODE SELECTION ALGORITHMS FOR H.264/AVC

Andy C. Yu

Department of Computer Science
University of Warwick, Coventry CV4 7AL, UK

Supervisor: Dr. Graham Martin
June 2004

EFFICIENT INTRA- AND INTER-MODE SELECTION ALGORITHMS FOR H.264/AVC

by

Andy C. Yu

ABSTRACT

H.264/AVC standard is one of the most popular video formats for the next generation video coding. It provides a better performance in compression capability and visual quality compared to any existing video coding standards. Intra-frame mode selection and inter-frame mode selection are new features introduced in the H.264/ AVC standard. Intra-frame mode selection dramatically reduces spatial redundancy in I-frames, while inter-frame mode selection significantly affects the output quality of P-/B-frames by selecting an optimal block size with motion vector(s) or a mode for each macroblock. Unfortunately, this feature requires a myriad amount of encoding time especially when a brute force full-search method is utilised.

In this report, we propose fast mode-selection algorithms tailored for both intra-frames and inter-frames. The proposed fast intra-frame mode algorithm is achieved by reducing the computational complexity of the Lagrangian rate-distortion optimisation evaluation. Two proposed fast inter-frame mode algorithms incorporate several robust and reliable predictive factors, including intrinsic complexity of the macroblock, mode knowledge from the previous frame(s), temporal similarity detection and the detection of different moving features within a macroblock, to effectively reduce the number of search operations. Complete and extensive simulations are provided respectively in these two chapters to demonstrate the performances.

Finally, we combine our contributions to form two novel fast mode algorithms for H.264/AVC video coding. The simulations on different classes of test sequences demonstrate a speed up in encoding time of up to 86% compared with the H.264/AVC benchmark. This is achieved without any significant degradation in picture quality and compression ratio.

Keywords: H.264/AVC, intra-frame mode selection, inter-frame mode selection, Lagrangian rate-distortion optimisation.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

Glossary

4:2:0 (sampling)	A colour sampling method. Chrominance components have only half resolution as luminance component (see Chapter 0).
AC	Alternative Current, refer to high frequency components.
ASVC	Advance Scalable Video Coding (see Chapter 5)
Arithmetic coding	A lossless coding method to reduce redundancy.
AVC	Advanced Video Coding (see Chapter 1)
Block	A region of a macroblock, normally 8x8 or 4x4 pixels.
Block matching	Motion estimation carried out on block-based.
CABAC	Context-based Adaptive Binary Arithmetic Coding.
CAVLC	Context-based Variable Length Coding.
CE	Core Experiment (see Chapter 5)
Chrominance	Colour space (see Chapter 0).
CIF	Common Intermediate Format (see Chapter 0).
CODEC	Coder / DECoder pair.
DC	Direct Current, refer to low frequency components.
DCT	Discrete Cosine Transform (see Chapter 0).

Entropy coding	A coding method make use of entropy (information of data), including Arithmetic coding and Huffman coding.
Finter1	Fast inter mode selection algorithm 1 (see Chapter 3).
Finter2	Fast inter mode selection algorithm 2 (see Chapter 3).
Fintra	Fast intra mode selection algorithm (see Chapter 2).
Full search	A motion estimation algorithm.
GOP	Group of Picture (see Chapter 0).
H.261	A video coding standard.
H.263	A video coding standard.
H.264/AVC	A video coding standard (see Chapter 1).
Huffman coding	An entropy coding method to reduce redundancy.
Inter (coding)	Coding of video frames using temporal block matching (see Chapter 0).
Intra (coding)	Coding of video frames without reference to any other frame (see Chapter 0).
I-picture/frame	Picture coded without reference to any other frame.
ISO	International Standard Organisation, a standards body.
ITU	International Telecommunication Union, a standards body.

JVT	Joint Video Team, collaboration team between ISO/IEC MPEG and ITU-T VCEG.
Macroblock	A basic building block of a frame/picture (see Chapter 0).
Motion compensation	Reconstruction of a video frame according to motion estimation of references (see Chapter 0).
Motion estimation	Prediction of relative motion between two or more video frames (see Chapter 0).
Motion vector	A vector indicates a displaced block or region to be used for motion compensation.
MPEG	Motion Picture Experts Group, a committee of ISO/IEC.
MPEG-1	A multimedia coding standard.
MPEG-2	A multimedia coding standard.
MPEG-4	A multimedia coding standard.
Objective quality	Visual quality measured by algorithm (see Chapter 0).
Picture/frame	Coded video frame.
P-picture/frame	coded picture/frame using motion-compensated prediction from one reference frame.
PSNR	Peak Signal to noise Ratio (see Chapter 0).
QCIF	Quarter Common Intermediate Format (see Chapter 0).

Quantise	Reduce the precision of a scalar of vector quantity (see Chapter 0).
RGB	Red/Green/Blue colour space (see Chapter 0).
SAD	Sum of Absolute Difference.
SVC	Scalable Video Coding.
Texture	Image or residual data.
VCEG	Video Coding Expert Group, a committee of ITU.
VLC	Variable Length Code.
YC_bC_r	Luminance, Blue chrominance, Red chrominance colour space (see Chapter 0).
YUV	a colour space (see Chapter 0).

Chapter 0
Video Basics

This chapter aims at defining several fundamentals on video basics and video compression. Those characteristics will help us to understand how video compression works without actually introducing perceptual distortion.

0.1 Colour components

Basically, three primary colour signals, red, green and blue signals (RGB-signal) are generated during the scanning of a video camera. However, the RGB-signal is not efficient for transmission and storage purposed because it occupies three times the capacity as a grey-scale signal. Due to high correlation among the three colour signals and compatibility with the grey-scale signal, NTSC, PAL, and SECAM standards [25] are generated to define the colour in different implementations. Among these, the PAL standard is widely used in video coding research to represent a colour video signal. It has three basic colour representations, YUV, where Y represents the luminance and U (or C_r) and V (or C_b) represent the two colour components. The conversion equations between RGB and YUV can be represented as

$$Y = 0.299R + 0.587G + 0.114B$$
$$U = 0.492(B - Y)$$
$$V = 0.877(R - Y)$$

0.2 Video format

Common Intermediate Format (CIF) and Quarter-CIF (QCIF) are two of the most popular formats for low bandwidth video applications. The standard CIF picture has luminance (Y) component with spatial resolution of 360 pixels per line and 288 lines

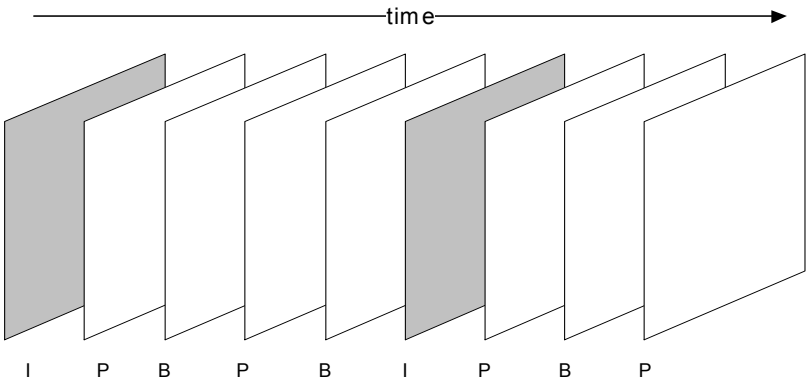

Fig. 0-1 Illustration of the three frame types (I-, P-, and B-).

per frame. The corresponding two chrominance (C_r and C_b) components have the same vertical resolution as luminance, but horizontal resolution is one-quarter. Such a combination of Y, U and V components is called the 4:2:0 sampling format. QCIF format, like the CIF format, makes use of the 4:2:0 sampling format but only one-quarter the size of CIF format.

0.3 The structure of a video sequence

A video sequence can be described as many groups of pictures (GOP) with three different types, Intra-coded (I-), Predictive (P-), and Bidirectional-predictive (B-) pictures/frames, Fig. 0-1. I-frames, the first frames in GOPs, are coded independently without any reference to other frames, whereas P- and B- frames are compressed by coding the differences between the picture and reference(s), either I- or other P-frames, thereby exploiting the redundancy from one frame to another.

Each frame of a video sequence is decomposed into smaller basic building blocks called macroblocks. A macroblock consists of a 16x16 sample array of luminance (Y) sample together with one 8x8 block of sample for each of two chrominance (C_r and C_b) components.

0.4 Motion Estimation and Compensation

Block-based motion estimation and compensation are used to exploit the temporal redundancies between the encoding frame and reference frame(s), Fig. 0-2. Motion compensation is a process of compensating for the displacement of moving objects

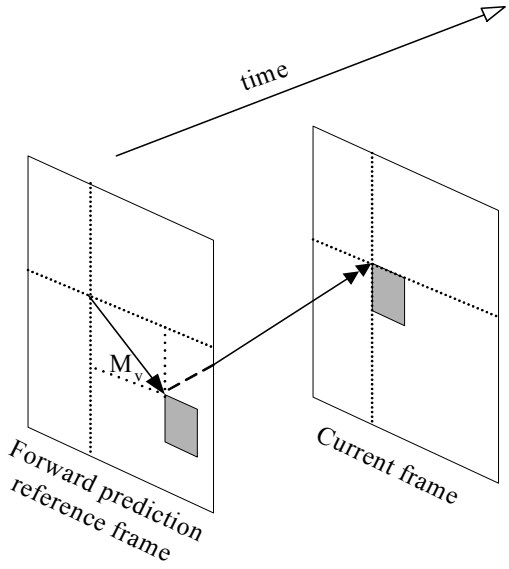

Fig 0-2 Motion compensated prediction and reconstruction.

from one frame to another. In practice, motion compensation is preceded by motion estimation, the process of finding a corresponding best matched block. In general, we segment the current frame into non-overlapping 16×16 pixel macroblocks, and for each macroblock, we determine a corresponding 16×16 pixel region in the reference frame. Using the corresponding 16×16 pixel region from the reference frame, the temporal redundancy reduction processor generates a representation for the current frame that contains only the changes between the two frames. If the two frames have a high degree of temporal redundancy, then the difference frame would have a large number of pixels that have values close to zero.

0.5 Transform coding

The function of block-based transform coding is to achieve energy compaction and separate the low spatial frequency information from high spatial frequencies. The discrete cosine transform is one of the most popular transformation methods utilised in video coding. The $N \times N$ two-dimensional DCT is defined as:

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

$$C(u,v) = \begin{cases} \frac{1}{\sqrt{2}} & \text{, for } u, v = 0 \\ 1 & \text{, otherwise} \end{cases}$$

where x, y are coordinates in the spatial domain,

and u, v are coordinates in the frequency domain.

0.6 Quantisation

Quantisation is an irreversible process to represent the coefficients for high spatial frequencies with less precision. That is because human perception is less sensitive to high spatial frequencies. A DCT coefficient is quantised/divided by a nonzero positive integer called a quantization value, q_{uv}, and the quotient, rounded to the nearest integer. The process of quantization, $Q(F(u,v))$, is expressed as

$$Q(F(u,v)) = \text{round}\left(\frac{F(u,v)}{q_{uv}}\right)$$

0.7 Visual quality evaluation

The most recognised objective measurement of visual quality is peak-to-peak signal-to-noise ratio (PSNR) [26]. It is defined as:

$$\text{PSNR} = 10\log_{10}\left[\frac{255 \times 255}{\frac{1}{M \times N}\sum_{i}^{M}\sum_{j}^{N}\left(\text{Y}_{\text{rec}}(i,j) - \text{Y}_{\text{ori}}(i,j)\right)^2}\right]$$

where $\text{Y}_{\text{rec}}(i,j)$ and $\text{Y}_{\text{ori}}(i,j)$ are the luminance values of the reconstructed and original video signals respectively, and M and N are the number of pixels in the horizontal and vertical directions.

0.8 Intra-coding and inter-coding

The working diagrams of the intra-coding and inter-coding processes are depicted in Fig. 0-3 and Fig. 0-4, respectively. These two block diagrams are similar to each other in terms of the DCT transformation, quantisation process, and entropy encoding. The difference is that the inter-frame encoder decomposes a video frame into several non-overlapped macroblocks (of size 16x16 pixels) rather than the 8x8 blocks in intra-coding. Each inter-macroblock has to undergo motion estimation to search the best matching blocks in the reference frame(s). Residue data is then obtained by subtracting the reconstructed frame (constructed from reference frame) from the original frame in the motion compensation process. Please note that only residue data is encoded in inter-frame coding, whereas intra-coding encodes all the pixel information.

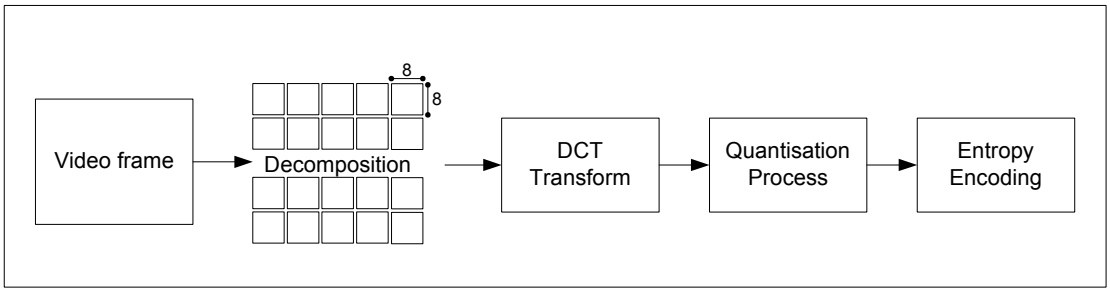

Fig. 0-3 Block diagram of intra-coding in general video compression.

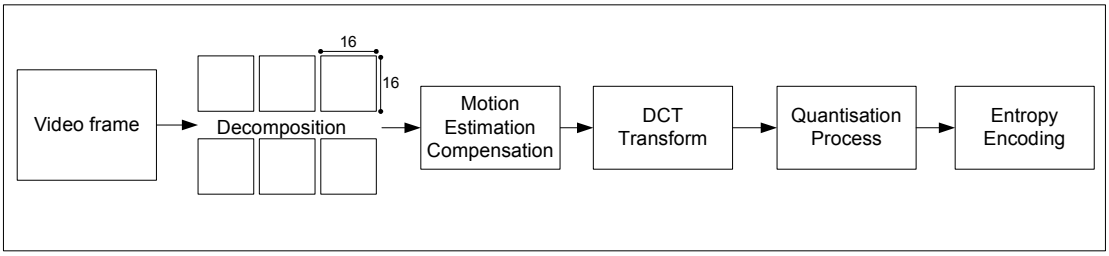

Fig. 0-4 Block diagram of inter-coding in general video compression.

Chapter 1
Overview of texture coding in H.264/AVC

1.1 Introduction

Moving Picture Experts Group (MPEG) is a working group in ISO/IEC, which has been playing pivotal role in establishing the international standards for video compression technologies. MPEG-1, MPEG-2, MPEG-4, MPEG-7, and MPEG-21 are five important standards identified by MPEG. In early 1998, Video Coding Expert Group (VCEG) in ITU-T SG16 Q.6 started a call for proposals on a project called H.26L, which is targeted to obtain a powerful video compression tool featuring high compression [24]. In July 2001, ITU-T called for technology and demonstrated the H.26L at MPEG ISO/IEC JTC1/SC29/WG11. Later, ISO/IEC MPEG and ITU-T VCEG decided to form a collaboration title Joint Video Team (JVT), which consists the experts from both organisations, in December 2001. The standard was renamed as H.264 by ITU-T, Advanced Video Coding (AVC, MPEG-4 part 10) by ISO/IEC [24].

The H.264/AVC is the latest and state-of-the-art video compression standard. The JVT experts addressed a number of advanced features of H.264/AVC [1]. These improvements achieve significant gains in encoder and decoder performances. One of the new features is multi-mode selection for intra-frames and inter-frames, which is the subject of this paper. In the H.264/AVC coding algorithm, block-matching motion estimation is an essential part of the encoder to reduce the temporal redundancy between two successive frames. The difference, however, is that the block size is no longer fixed. The block size is variable ranging from 4×4 to 16×16 [1] in inter-frame coding (Fig. 1-1), in order to minimise the overall prediction error. Furthermore, intra-frame modes, where the objective is to reduce the spatial

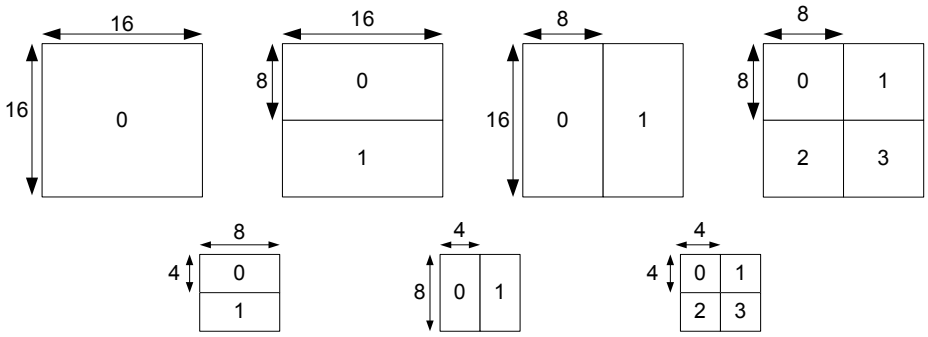

Fig. 1-1 *INTER* modes with 7 different block sizes ranging from 4×4 to 16×16.

redundancy in a frame, constitute the other candidates for mode selection. The effect is to increase the complexity of the mode-selection scheme.

1.2 Lagrangian rate-distortion optimisation

The method employed by the H.264/AVC standard to make a mode decision requires the application of Lagrangian rate-distortion optimisation. The optimisation approach is based on the assumption that the distortion and rate incurred in coding a macroblock are independent of each other [3]. Hence, the coding mode of each macroblock is acquired from knowledge of the previously coded blocks. Let us denote $\mathbf{B}_t$ as a block of any rectangular size in a frame at time t; while $\hat{\mathbf{B}}_{t-\tau}$ is a reconstructed block of the same block size as $\mathbf{B}_t$ located in the previously coded frame at time $t-\tau$ ($\tau = 0$ in intra-frame coding). Then, the macroblock-based Lagrangian cost LC_{MB} for $\mathbf{B}_t$ is:

$$LC_{MB}(\mathbf{B}_t, \hat{\mathbf{B}}_{t-\tau}, \text{mode} \mid Qp, \lambda_{\text{mode}}) = D(\mathbf{B}_t, \hat{\mathbf{B}}_{t-\tau}, \text{mode} \mid Qp) + \lambda_{\text{mode}} \bullet R(\mathbf{B}_t, \hat{\mathbf{B}}_{t-\tau}, \text{mode} \mid Qp) \quad (1)$$

where Qp and λ_{mode} represent the macroblock quantiser value and Lagrange parameter, respectively. λ_{mode} is normally associated with Qp and has a relationship

approximated as $0.85\times Qp^2$ [3-6]. In the H.264/AVC standard, the alternative definition for λ_{mode} is:

$$\lambda_{\text{mode}} = 5 \bullet e^{Qp/10} \bullet \left(\frac{Qp+5}{34-Qp} \right). \tag{2}$$

In (1), D is a distortion measure quantifying the difference between $\mathbf{B}_t$ and $\hat{\mathbf{B}}_{t-\tau}$, defined separately in terms of intra- and inter-frame mode as:

$$D(\mathbf{B}_t, \hat{\mathbf{B}}_t, \text{intra mode} \,|\, Qp) = \sum_x \sum_y \left| \mathbf{B}_t(x,y) - \hat{\mathbf{B}}_t(x,y,\text{mode} \,|\, Qp) \right|^p , \tag{3a}$$

$$D(\mathbf{B}_t, \hat{\mathbf{B}}_{t-i}, \text{inter mode} \,|\, Qp) = \sum_x \sum_y \left| \mathbf{B}_t(x,y) - \hat{\mathbf{B}}_{t-\tau}(x+m_x, y+m_y, \text{mode} \,|\, Qp) \right|^p , \tag{3b}$$

where (m_x, m_y) represents the motion vector in the inter-frame case.

R in (1) reflects the number of bits associated with choosing the mode and Qp including the bits for the macroblock header, the motion vector(s) and all the DCT residue blocks. It can be obtained from the look-up table of run-level variable-length codes. Mode indicates a mode chosen from the set of potential prediction modes, the respective possibilities of which are:

$$\text{mode}_{\text{intra}} \in \{I4MB, I16MB\}, \tag{4}$$

$$\text{mode}_{\text{inter}} \in \{SKIP, I4MB, I16MB, INTER\}. \tag{5}$$

Intra-frame mode has two modes, *I4MB* and *I16MB*. *I4MB* consists of 9 members which pad elements (*a* to *p*) of a 4×4 block with the neighbouring encoded pixels (*A* to *Q*) in 8 directions as depicted in Fig. 1-2 and Fig. 1-3, respectively. For instance, *VERT*, the vertical member, pads a 4×4 block vertically with 4 neighbouring pixels, *A*, *B*, *C*, *D*, whereas the horizontal member, *HORT*, utilizes the horizontal adjacent pixels, *I*, *J*, *K*, *L* to do the prediction. The other modes operate the same way according to their corresponding orientations, except for *DC*, the directionless member, which pads all pixels with (*A*+*B*+*C*+*D*+*I*+*J*+*K*+*L*)/8. *I16MB* resembles *I4MB* but is less time-

consuming, comprising 4 members to predict a 16×16 macroblock as a whole. As for inter-frame mode, it contains the *SKIP* (direct copy), *I4MB*, *I16MB*, and *INTER*, the most time-consuming mode, which consists of 7 members with different block sizes as shown in Fig. 1-1.

In intra-frame coding, the final mode decision is selected by the member (either from *I4MB* or *I16MB*) that minimizes the Lagrangian cost in (1). In inter-frame coding, motion estimations with 7 different block-size patterns, as well as the other members in three members (*I4MB*, *I16MB*, and *SKIP*), are calculated. The final decision is determined by the mode that produces the least Lagrangian cost among the available modes. Currently, the H.264/AVC standard employs a brute force algorithm to search through all the possible candidates and its corresponding members to find an

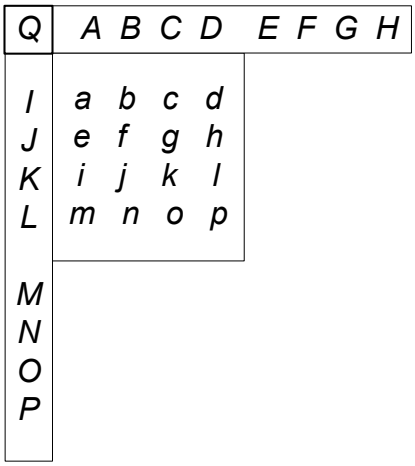

Fig. 1-2 A 4×4 block with elements (a to p) which are predicted by its neighbouring pixels.

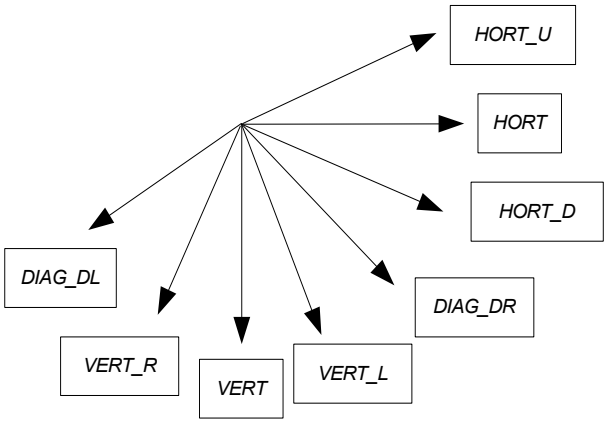

Fig. 1-3 Eight direction-biased *I4MB* members except *DC* member which is directionless.

optimum motion vector [2]. Since the exhaustive search method is employed in all the modes to acquire a final mode decision, the computational burden of the search process is far more significant than any existing video coding algorithm.

1.3 Contributions and organisation of this report

The contributions of this report are to develop fast mode selection algorithms to reduce the computational time for both intra- and inter-frame coding. The proposed algorithm comprises two parts: (1) fast intra-frame mode selection (*Fintra*) algorithm designed to acquire the most likely prediction modes from the *I4MB* mode from knowledge of the frequency spectrum; (2) two fast inter-frame mode selection algorithms (denoted as *Finter1* and *Finter2*). The next two chapters give detailed formulation of the proposed algorithms. The simulation results of two combined algorithms (*Finter1* + *Fintra* and *Finter2* + *Fintra*) are summarized in Chapter 4. Finally, Chapter 5 discusses some overall conclusions and prospective projects in coming two years.

Chapter 2
Proposed intra-frame mode selection algorithm

2.1 Introduction

In intra-frame coding, the H.264/AVC standard selects a mode which minimizes the Lagrangian cost LC_{MB} as given in (1). The optimisation process entails finding the least distortion while achieving the minimum coding rate. The computation of the distortion parameter, D, requires the availability of the reconstructed image, which means the completion of the encoding-decoding cycle. On the other hand, the evaluation of the rate parameter, R, depends only on the residue blocks obtained from the difference between the original block and the predicted block for each mode by look-up table of the entropy codes. Clearly, the computational requirement of rate evaluation is less demanding than that for the distortion evaluation.

It is observed that the modes that provide the least residue energy will also result in minimum rate R and hence minimise the Lagrangian cost. The chart in Fig. 2-1 illustrates this observation by showing the match percentage between the candidate(s) with least distortion cost and the mode decision with least rate-distortion cost acquired with Lagragian evaluation in (1). Thirty frame of three test sequences in CIF (352×288 pels) resolution, *Akiyo* (Class A), *Foreman* (Class B), and *Mobile & Calendar* (Class C), were intra-coded in order to obtain the results. The first bar of each test sequence represents the match score between the mode decision and the *MostProbableMode*, the candidate mode predicted from use of prior knowledge of neighbouring blocks. It shows that the match percentages in the three test sequences are 56%, 42%, and 30%. However, the match percentages surge when the number of candidates with the least distortion cost increases. The simulation results shows a

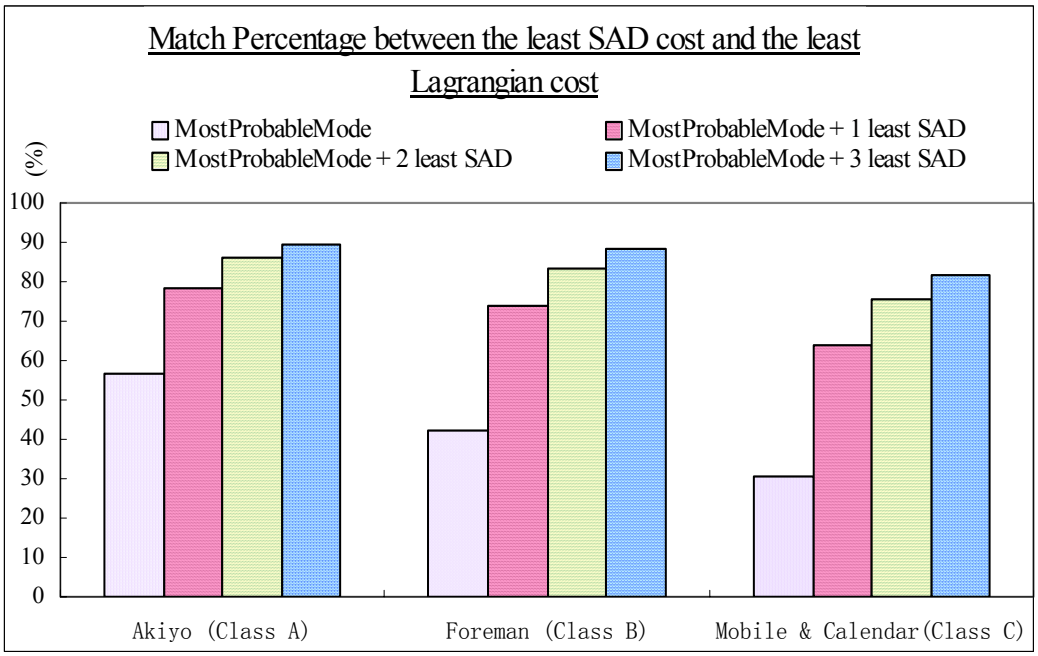

Fig. 2-1 Match percentage between the least distortion cost acquired from SAD implementation and the least rate-distortion cost obtained from Lagrangian evaluation.

match percentage of 89%, 88% and 81% for the three respective test sequences when the number of candidates increases to 4 including *MostProbableMode*.

Therefore, to reduce the computational cost of the expensive Lagrangian cost evaluation, we can limit the number of members (say *M*) that need to undergo the full evaluation process. The *M* members are those with the least residue energy from amongst all the possible members. Furthermore, the residue blocks of *I4MB* and *I16MB* normally have relatively large block energy because there is no prediction. Hence, it is more efficient to operate in the frequency domain rather than in the spatial domain. The following subsections detail the formulation of the fast algorithm.

2.2 Algorithm formulation

The proposed fast intra-mode selection (*Fintra*) is achieved by selecting fewer members from *I4MB* mode that need to undergo the full Lagrangian cost evaluation. The selection criterion is the least residue energy which can be measured from the sum of absolute difference (SAD) of the DCT residue block. First, let us denote an

$M \times N$ original block to be $\mathbf{B}_{M \times N}$ and any intra predicted block to be $\mathbf{P}_{M \times N,\ \text{member}}$. For a unitary transform, the SAD of the DCT residue block is given by

$$\text{SAD}_{\text{DCT(residue)}} = \sum \left| Diff(T\{\mathbf{B}_{M \times N}\}, T\{\mathbf{P}_{M \times N,\ \text{mode}}\}) \right| = \sum \left| T\{Diff(\mathbf{B}_{M \times N}, \mathbf{P}_{M \times N,\ \text{member}})\} \right| \quad (6)$$

where *Diff*(**A**,**B**) represents the difference between **A** and **B**, where $T\{.\}$ stands for the unitary transformation. In our case, $T\{.\}$ stands for the Discrete Cosine Transform (DCT). From (6), a SAD evaluation is equal to the sum of absolute difference between the transforms of an original DCT-block, $T\{\mathbf{B}_{M \times N}\}$ and a predicted DCT-block, $T\{\mathbf{P}_{M \times N,\ \text{member}}\}$. Then, according to the definition of DCT,

$$\sum \left| Diff(T\{\mathbf{B}_{M \times N}\}, T\{\mathbf{P}_{M \times N,\ \text{member}}\}) \right| = \left| DC_{\mathbf{B}} - DC_{\mathbf{P},\ \text{member}} \right| + \sum \left| AC_{\mathbf{B}} - AC_{\mathbf{P},\ \text{member}} \right| \quad (7)$$

Equation (7) indicates that $\text{SAD}_{\text{DCT(residue)}}$ can be obtained by finding the sum of the absolute differences of both the low-frequency (DC) coefficients and the high-frequency (AC) coefficients. Note that a DC coefficient normally possesses more block energy than the AC coefficients for natural images. Thus, we can formulate the approximation as:

$$\left| T\{Diff(\mathbf{B}_{M \times N}, \mathbf{P}_{M \times N,\ \text{member}})\} \right| \approx \left| DC_{\mathbf{B}} - DC_{\mathbf{P},\ \text{member}} \right| + \left| AC'_{\mathbf{B}} - AC'_{\mathbf{P},\ \text{member}} \right|, \quad (8)$$

where $AC'_{\mathbf{B}}$ represents the AC coefficient that possesses the largest energy of these AC coefficients in an original DCT-block, and $AC'_{\mathbf{P},\ \text{member}}$ is the AC coefficient that is at the same location as $AC'_{\mathbf{B}}$ in any predicted DCT-block. Since empirical experiments show that the low-frequency AC coefficients contain more energy than the high-frequency coefficients, we select $AC'_{\mathbf{B}}$ from the lower horizontal and vertical frequencies, for example, *AC(0,1)*, *AC(0,2)*, *AC(0,3)* and *AC(1,0)*, *AC(2,0)*, *AC(3,0)*, as the candidates in a 4×4 block. By simple calculations from the 2D-DCT definition, we can easily obtain the formulae of the following $AC'_{\mathbf{B}}$ candidates of a 4×4 block:

$$DC_{\mathbf{B}} = f_0 \times \sum\sum \mathbf{B}_{4\times 4,\, t}\,, \tag{9}$$

$$AC_{\mathbf{B}}(1,0) = f_1 \times [r_{\mathbf{B}}(0) - r_{\mathbf{B}}(3)] + f_2 \times [r_{\mathbf{B}}(1) - r_{\mathbf{B}}(2)], \tag{10}$$

$$AC_{\mathbf{B}}(0,1) = f_1 \times [c_{\mathbf{B}}(0) - c_{\mathbf{B}}(3)] + f_2 \times [c_{\mathbf{B}}(1) - c_{\mathbf{B}}(2)], \tag{11}$$

$$AC_{\mathbf{B}}(2,0) = f_0 \times [r_{\mathbf{B}}(0) - r_{\mathbf{B}}(1) - r_{\mathbf{B}}(2) + r_{\mathbf{B}}(3)], \tag{12}$$

$$AC_{\mathbf{B}}(0,2) = f_0 \times [c_{\mathbf{B}}(0) - c_{\mathbf{B}}(1) - c_{\mathbf{B}}(2) + c_{\mathbf{B}}(3)], \tag{13}$$

$$AC_{\mathbf{B}}(3,0) = f_2 \times [r_{\mathbf{B}}(0) - r_{\mathbf{B}}(3)] + f_1 \times [r_{\mathbf{B}}(1) - r_{\mathbf{B}}(2)], \tag{14}$$

$$AC_{\mathbf{B}}(0,3) = f_2 \times [c_{\mathbf{B}}(0) - c_{\mathbf{B}}(3)] + f_1 \times [c_{\mathbf{B}}(1) - c_{\mathbf{B}}(2)], \tag{15}$$

where f_0, f_1, f_2 are scalars and the values are 0.2500, 0.3267 and 0.1353, respectively. $r_{\mathbf{B}}(m)$ and $c_{\mathbf{B}}(n)$ represent the sum of the image intensities in the m^{th} row and n^{th} column of $\mathbf{B}_{4\times 4}$, respectively (refer to Fig. 2). For example, $r_{\mathbf{B}}(0) = a + b + c + d$.

Next, we consider how to efficiently access the $DC_{\mathbf{P},\,\text{member}}$ and $AC'_{\mathbf{P},\,\text{member}}$ values of the predicted block, $\mathbf{P}_{4\times 4,\,\text{member}}$. Unlike the original block, the predicted blocks are the direction-biased paddings from the neighbouring pixels. In order to simplify the calculation, we rewrite each of the equations of (9) to (15) in matrix form, i.e.,

$$AC'_{\mathbf{P},\,\text{member}} = \Theta_{\text{POS}(AC'_{\mathbf{B}})} \bullet \begin{bmatrix} a \\ \vdots \\ \vdots \\ p \end{bmatrix}, \tag{16}$$

where POS($AC'_{\mathbf{B}}$) stands for position of $AC'_{\mathbf{B}}$ and $\Theta_{\text{POS}(AC'_{\mathbf{B}})} = [\theta_1, \theta_2, \ldots, \theta_{16}]$ is a time-frequency conversion transpose vector between an AC'_P and the predicted elements (a to p). For instance, if POS($AC'_{\mathbf{B}}$) is selected at (2,0), then according to (12)

$$\Theta_{(2,0)} = [\underbrace{f_0,..,f_0}_{4}, \underbrace{-f_0,.....,-f_0}_{8}, \underbrace{f_0,..,f_0}_{4}]. \tag{17}$$

In a similar manner, a matrix formula can be provided to relate the predicted elements and the neighbouring samples (*A* to *Q*):

$$\begin{bmatrix} a \\ \vdots \\ \vdots \\ p \end{bmatrix} = \mathbf{C}_{\text{member}} \bullet \begin{bmatrix} A \\ \vdots \\ \vdots \\ Q \end{bmatrix} = \begin{bmatrix} C_{1,1} & C_{1,2} & \cdots & C_{1,16} & C_{1,17} \\ C_{2,1} & \ddots & & & C_{2,17} \\ \vdots & & \ddots & & \vdots \\ C_{15,1} & & & \ddots & C_{15,17} \\ C_{16,1} & C_{16,2} & \cdots & C_{16,16} & C_{16,17} \end{bmatrix} \begin{bmatrix} A \\ \vdots \\ \vdots \\ Q \end{bmatrix}, \tag{18}$$

where $\mathbf{C}_{\text{member}}$ is a 16-by-17 conversion matrix, for instance, $\mathbf{C}_{\text{HORT}}$, the conversion matrix of the horizontal member, pads the horizontal pixel *I* to the first row's elements, i.e., *a* to *d*. Then, all the coefficients in the first 4 rows of $\mathbf{C}_{\text{HORT}}$ are zero except for the ninth coefficients ($\mathbf{C}_{1,9}$, $\mathbf{C}_{2,9}$, $\mathbf{C}_{3,9}$, $\mathbf{C}_{4,9}$, i.e., position of *I*), which are one.

We then obtain the relationship between $AC'_{\mathbf{P},\text{member}}$ and the neighbouring pixels (*A* to *Q*) by combining (16) and (18).

$$AC'_{\mathbf{P}} = \boldsymbol{\omega}_{\text{member},\text{POS}(AC'_{\mathbf{B}})} \bullet \begin{bmatrix} A \\ \vdots \\ \vdots \\ Q \end{bmatrix}, \tag{19}$$

where $\boldsymbol{\omega}_{\text{member},\text{POS}(AC'_{\mathbf{B}})}$ is a 1-by-17 transpose vector. By arranging the elements of $\boldsymbol{\omega}_{\text{member},\text{POS}(AC'_{\mathbf{B}})}$ to form a matrix, we can obtain the values of $AC'_{\mathbf{P},\text{member}}$ for all the nine *I4MB* members.

$$\begin{bmatrix} AC'_{\mathbf{P},\text{member}1} \\ AC'_{\mathbf{P},\text{member}2} \\ \vdots \\ AC'_{\mathbf{P},\text{member}9} \end{bmatrix} = \boldsymbol{\Omega}_{\text{POS}(AC'_{\mathbf{B}})} \bullet \begin{bmatrix} A \\ \vdots \\ \vdots \\ Q \end{bmatrix}, \tag{20}$$

where

$$\mathbf{\Omega}_{\mathrm{POS}(AC'_{\mathbf{B}})} = \begin{bmatrix} \boldsymbol{\omega}_{\mathrm{member\,1,POS}(AC'_{\mathbf{B}})} \\ \boldsymbol{\omega}_{\mathrm{member\,2,POS}(AC'_{\mathbf{B}})} \\ \vdots \\ \boldsymbol{\omega}_{\mathrm{member\,9,POS}(AC'_{\mathbf{B}})} \end{bmatrix}. \tag{21}$$

$\mathbf{\Omega}_{\mathrm{POS}(AC'_{\mathbf{B}})}$ in (21) is a 9-by-17 sparse matrix. Similarly, $\mathbf{\Omega}_{DC}$ exists to obtain the values of $DC_{\mathbf{P},\,\mathrm{member}}$ for the 9 prediction modes.

$$\begin{bmatrix} DC'_{\mathbf{P},\,\mathrm{member\,1}} \\ DC'_{\mathbf{P},\,\mathrm{member\,2}} \\ \vdots \\ DC'_{\mathbf{P},\,\mathrm{member\,9}} \end{bmatrix} = \mathbf{\Omega}_{DC} \bullet \begin{bmatrix} A \\ \vdots \\ \vdots \\ Q \end{bmatrix}, \tag{22}$$

and $\mathbf{\Omega}_{DC}$ can be deduced in a similar manner from (9), (16), and (18).

$$\mathbf{\Omega}_{DC} = \begin{bmatrix} \boldsymbol{\omega}_{\mathrm{member\,1},DC} \\ \boldsymbol{\omega}_{\mathrm{member\,2},DC} \\ \vdots \\ \boldsymbol{\omega}_{\mathrm{member\,9},DC} \end{bmatrix}, \tag{23}$$

where,

$$\boldsymbol{\omega}_{\mathrm{member},DC} = \begin{bmatrix} f_0 \times C_{1,1} & f_0 \times C_{1,2} & \cdots & f_0 \times C_{1,16} & f_0 \times C_{1,17} \\ f_0 \times C_{2,1} & \ddots & & & f_0 \times C_{2,17} \\ \vdots & & \ddots & & \vdots \\ f_0 \times_0 C_{15,1} & & & \ddots & f_0 \times C_{15,17} \\ f_0 \times C_{16,1} & f_0 \times C_{16,2} & \cdots & f_0 \times C_{16,16} & f_0 \times C_{16,17} \end{bmatrix}. \tag{24}$$

Note that $\mathbf{\Omega}_{DC}$ and all six $\mathbf{\Omega}_{\mathrm{POS}(AC'_{\mathbf{B}})}$ can be calculated and stored in advance.

2.3 Proposed fast Algorithm

The proposed *Fintra* algorithm utilizes (20) and (22) to shortlist M (<9) candidates from the 9 prediction modes. However, since empirical trials indicate that

MostProbableMode (the mode predicted from use of prior knowledge of neighbouring blocks) has a higher chance of being selected as the prediction mode, it is included in the short-listed candidates although it may not produce the least residue energy. The proposed algorithm is summarized as follows:

A1. Evaluate (9)-(15) to obtain $DC_{\mathbf{B}}$ and an $AC'_{\mathbf{B}}$, the AC coefficient which possesses the largest AC energy of the original block.
A2. Calculate values of $DC_{\mathbf{P},\text{member}}$ and $AC'_{\mathbf{P},\text{member}}$ of the 9 predicted blocks by utilizing (22) and (20).
A3. Apply SAD evaluation in (8) to shortlist 1-4 candidates with the smallest residue energies (including *MostProbableMode*).
A4. Select a prediction mode that minimizes (1) from the short-listed candidates.

The proposed intra-frame mode selection algorithm, *Fintra*, employs the inherent frequency characteristic of an original block and its predicted block without any a priori knowledge, such as predefined threshold or other priori macroblock information. This feature is considered one of the main advantages of the proposed algorithm in that it can be easily applied to the *I16MB* and mode selection for chrominance components from one sequence to another. Furthermore, the matrices, all $\mathbf{\Omega}_{\text{POS}(AC'_{\mathbf{B}})}$ in different $AC'_{\mathbf{B}}$ positions and $\mathbf{\Omega}_{DC}$, can be calculated and stored in advance.

2.4 Simulation results

All the simulations presented in this section were programmed using C++. The computer used for the simulations was a 2.8GHz Pentium 4 with 1024MB RAM. The testing benchmark was the JM6.1e version provided by the Joint Video Team (JVT) [12]. The selected sequences in 2 different resolutions, namely, QCIF (144×176) and CIF (288×352) formats, are classified into three different classes, i.e., Class A, which are sequences containing only low spatial correlation and motion, e.g., *Akiyo* and *Ship Container*; Class B, which contains medium spatial correction and/or motion, e.g., *Foreman* and *Silent Voice*; and Class C, where high spatial correlation and/or motion are involved, e.g., *Mobile & Calendar* and *Stefan*. The other settings are as follows:

all sequences were quantized by a static *Qp* factor of 32. They were encoded by the intra-coding technique provided by JM6.1e and the proposed *Fintra* algorithm. In each case, the rate was 30 frames per second with no skip frame throughout the 30 frames.

TABLE 2-1
Simulation results of the proposed *Fintra* algorithm compared with JM6.1e, the H.264/AVC software, in three sequence classes and two resolutions.

Classes / Sequences		Resolutions (pels)	Y-PSNR Difference (dB)	Bit Rate Difference (%)	Speed-up cf.JM6.1e (%)
A	Akiyo	144×176	-0.01 dB	0.24%	51.85%
	Grandma	144×176	-0.02 dB	0.50%	51.59%
	Hall Monitor	288×352	-0.02 dB	0.33%	59.40%
	Mother & Daughter	288×352	-0.01 dB	0.31%	57.98%
B	City	288×352	-0.04 dB	0.19%	54.65%
	Coastguard	144×176	-0.03 dB	-0.18%	54.95%
	Foreman	288×352	-0.02 dB	0.26%	59.13%
	News	144×176	-0.04 dB	0.28%	53.90%
	Paris	288×352	-0.04 dB	0.21%	59.97%
C	Car Phone	144×176	-0.01 dB	0.49%	51.75%
	Crew	288×352	-0.01 dB	0.21%	55.10%
	Harbour	288×352	-0.03 dB	0.14%	61.50%
	Football	144×176	-0.05 dB	0.43%	52.42%
	Mobile & Calendar	288×352	-0.09 dB	0.16%	55.17%
	Table Tennis	288×352	-0.01 dB	0.01%	51.92%
	Waterfall	288×352	-0.04 dB	0.03%	55.05%

Table 2-1 shows the simulation results of the proposed *Fintra* algorithm in comparison with the JM6.1e implementation. Comparisons are given for PSNR difference in the luminance component, Y-PSNR (measured in dB), bit rate difference (as a percentage), and speedup (computational performance). The Table 2 entries are arranged according to class of sequence.

The general trends are identified as follows: all the selected sequences are able to attain almost the same PSNR performance and bit rates as the JM6.1e algorithm. The selected sequences from Class A and Class B exhibit marginal PSNR differences which are between 0.01dB and 0.04dB, whereas sequences of Class C have a slightly wider range from 0.01dB to 0.09dB. As for time efficiency, it varies insignificantly among the test sequences. This is because the saving in time was achieved by

reducing the short-listed candidates for each block (see algorithm A4) regardless of the resolution and class of test sequences. On average, more than 50% of the encoding time is saved when the proposed *Fintra* algorithm is applied; the saving can be up to 62%.

Chapter 3
Proposed inter-frame mode selection algorithms

3.1 Introduction

Success of two proposed fast mode selection algorithms, *Finter1* and *Finter2*, for inter-frame coding is achieved by discarding the least possible block size. Mode knowledge of the previously encoded frame(s) is employed by the proposed *Finter1* algorithm, whereas the *Finter2* algorithm incorporates temporal similarity detection and the detection of different moving features within a macroblock. However, both *Finter1* and *Finter2* make use of a general tendency: a mode having a smaller partition size may be beneficial for detailed areas during the motion estimation process, whereas a larger partition size is more suitable for homogeneous areas [7]. Therefore the primary goal is to determine a complexity measurement for each macroblock.

3.2 Algorithm formulation

In this subsection, we derive a low-cost complexity measurement based on summing the total energy of the AC coefficients to estimate the block detail. The AC coefficients are obtained from the DCT coefficients of each block. The definition is

$$E_{AC} = \sum_{u \cup v \neq 0}^{M-1,\, N-1} \left(F_{uv}^2 \right), \tag{25}$$

where,

$$F_{uv} = c(u)c(v)\sum_{n=0}^{N-1}\sum_{m=0}^{M-1}\left[I_{mn}\cos\left(\frac{(2m+1)u\pi}{16}\right)\cos\left(\frac{(2n+1)v\pi}{16}\right)\right], \tag{26}$$

and,

$$c(u), c(v) = \begin{cases} \sqrt{\frac{1}{M}}, \sqrt{\frac{1}{N}} & for\ u, v = 0 \\ \sqrt{\frac{2}{M}}, \sqrt{\frac{2}{N}} & for\ u, v \neq 0 \end{cases}, \tag{27}$$

and I_{mn} stands for the luminance intensity located at (m, n) of an $M \times N$ block.

From (25), the total energy of the AC components, E_{AC}, of an $M \times N$ block is the sum of all the DCT coefficients, F_{uv}, except for the DC component, $u = 0$ and $v = 0$. According to the energy conservation principle, the total energy of an $M \times N$ block is equal to the accumulated energy of its DCT coefficients. Thus, (25) can be further simplified as

$$E_{AC} = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\left(I_{mn}^2\right) - \frac{1}{M}\frac{1}{N}\left(\sum_{n=0}^{N-1}\sum_{m=0}^{M-1} I_{mn}\right)^2, \tag{28}$$

where the first term is the total energy of the luminance intensities within an $M \times N$ block, and the second term represents the mean square intensity. (28) clearly shows that the energy of the AC components of a macroblock can be represented by the variance.

Since complexity measurements for different block sizes need to be made for each macroblock (up to 21 measurements per macroblock in the worst case), equation (28) can be further modified to form three piecewise equations to reduce the computational redundancy.

$$E_{AC} = \begin{cases} \sum_{x=1}^{16} E_x - \left(\frac{1}{16} \sum_{x=1}^{16} S_x \right)^2 & \text{(29 - a)} \\ \sum_{x=1}^{4} E_{4(n-1)+x} - \left(\frac{1}{8} \sum_{x=1}^{16} S_{4(n-1)+x} \right)^2, \quad n = \{1, \cdots, 4\} & \text{(29 - b)} \\ E_x - \left(\frac{1}{4} S_x \right)^2, \quad x = \{1, \cdots, 16\} & \text{(29 - c)} \end{cases}$$

where $E_n=\{e_1, e_2, \ldots, e_{16}\}$ and $S_n=\{s_1, s_2, \ldots, s_{16}\}$ represent the sum of energies and intensities of the 4×4 blocks decomposed from a macroblock respectively, with the scanning pattern shown in Fig. 3-1. The first piecewise equation is applied to a macroblock with block size of 16×16 pixels; the second is for 4 blocks, n = {1, 2, 3, 4} of 8×8 pixels; and the last is applicable to the 16 decomposed 4×4 blocks.

Evaluating the maximum sum of the AC components is the next target. By definition, the largest variance is obtained from the block comprising a checkerboard pattern in which every adjacent pixel is the permissible maximum (I_{max}) and minimum (I_{min}) value alternately [8]. Thus, E_{max}, the maximum sum of AC components of an $M \times N$ block is

$$E_{\max} = \left[\left(I_{\max}^2 + I_{\min}^2 \right) - \frac{1}{2} \left(I_{\max} + I_{\min} \right)^2 \right] \times \frac{M \times N}{2}. \tag{30}$$

Note that E_{max} can be calculated in advance. Then the criterion to assess the complexity R_B of a macroblock MB is

$$R_B = \frac{\ln(E_{AC})}{\ln(E_{\max})}. \tag{31}$$

The function of the natural logarithm is to linearise both E_{max} and E_{AC} such that the range of R_B can be uniformly split into 10 subgroups. In our evaluation, a macroblock with $R_B > 0.75$ is considered to be a high-detailed block.

3.3 The proposed *Finter1* algorithm

Fig. 3-2 shows the flowchart of the proposed *Finter1* algorithm that incorporates the complexity measurement. In total, 7 partition sizes are recommended by H.264/AVC for P-frames, namely, 16×16, 16×8, 8×16, 8×8, 8×4, 4×8, 4×4 as well as *SKIP*, *4MB* and *16MB*. However, in our complexity measurement, only 3 categories, of sizes of 16×16, 8×8, and 4×4, respectively, are selected as test block sizes. We denote them as *Cat0*, *Cat1*, and *Cat2*, respectively.

The proposed *Finter1* algorithm provides a recursive way to determine the complexity of each macroblock. Firstly, a macroblock of 16×16 pixels is examined with (29-a). A *Cat0* tag is given if it is recognized as being a homogenous macroblock. Otherwise, the macroblock is decomposed into 4 blocks of 8×8 pixels. Note that an 8×8 block is recognized as high-detailed if it satisfies two conditions: (a) the R_B in (31) is greater than 0.75, and it is decomposed into four 4×4 blocks, and (b) one of its four decomposed 4×4 blocks is high-detailed as well. If an 8×8 block satisfies the first condition but not the second, it is still recognized as low-detailed. After checking all the 8×8 blocks, a *Cat2* tag is given to a macroblock which possesses more than two high-detailed blocks, otherwise a *Cat1* tag is assigned. Table 3-1 displays the relationship between the three categories in the proposed algorithm and the nine members of the inter-frame modes. It is observed that the *Cat0* category covers the

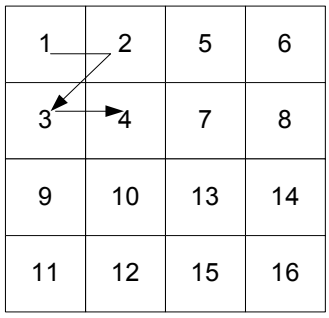

Fig. 3-1 The proposed scanning order of E_n and S_n, the energy and sum of intensities in 4×4 block in order to reduce computational redundancy.

least number of members of the inter-frame mode, whereas the *Cat2* category contains all the available members. The table further indicates that the higher detailed the macroblocks are, the more prediction modes the proposed algorithm has to check.

TABLE 3-1
The relationship between the three categories in the proposed algorithm and the 9 members of inter-frame modes.

Category	Corresponding Modes
Cat0	16×16 , *SKIP*, *I16MB*, *I4MB*
Cat1	16×16 , 16×8 , 8×16 , 8×8 , *SKIP*, *I16MB*, *I4MB*
Cat2	16×16 , 16×8 , 8×16 , 8×8 , 8×4 , 4×8 , 4×4 , *SKIP*, *I16MB*, *I4MB*

Mode knowledge of previously encoded frame(s):

A trade-off between efficiency and prediction accuracy exists. If a *Cat2* category is assigned less often, the efficiency of the algorithm will increase, but the chance of erroneous prediction also increases. An improved method is proposed, that considers the mode knowledge at the same location in the previously encoded frame. Since most of the macroblocks are correlated temporally, it is easy to see that the mode decision in the previous frame contributes reliable information for revising the erroneous prediction that may be indicated by its intrinsic complexity information. Therefore, our suggestion is first to convert all the mode decisions in the previous frame into the corresponding categories. Then, the prediction is revised to the higher category if that of the corresponding historic data is higher than the current predictor. However, no action is taken if the reverse situation is true.

The algorithm of *Finter1*:

Cat0 category algorithm:

B1. Obtain a motion vector for a 16×16 macroblock by using the full search algorithm with search range of ± 8 pixels.

B2. The best prediction of *I4MB* and *I16MB* can be obtained by applying steps A1 to A4 and the full search algorithm, respectively.

B3. Compute the Lagrangian costs of *SKIP*, *I4MB*, *I16MB*, and *INTER* to find a

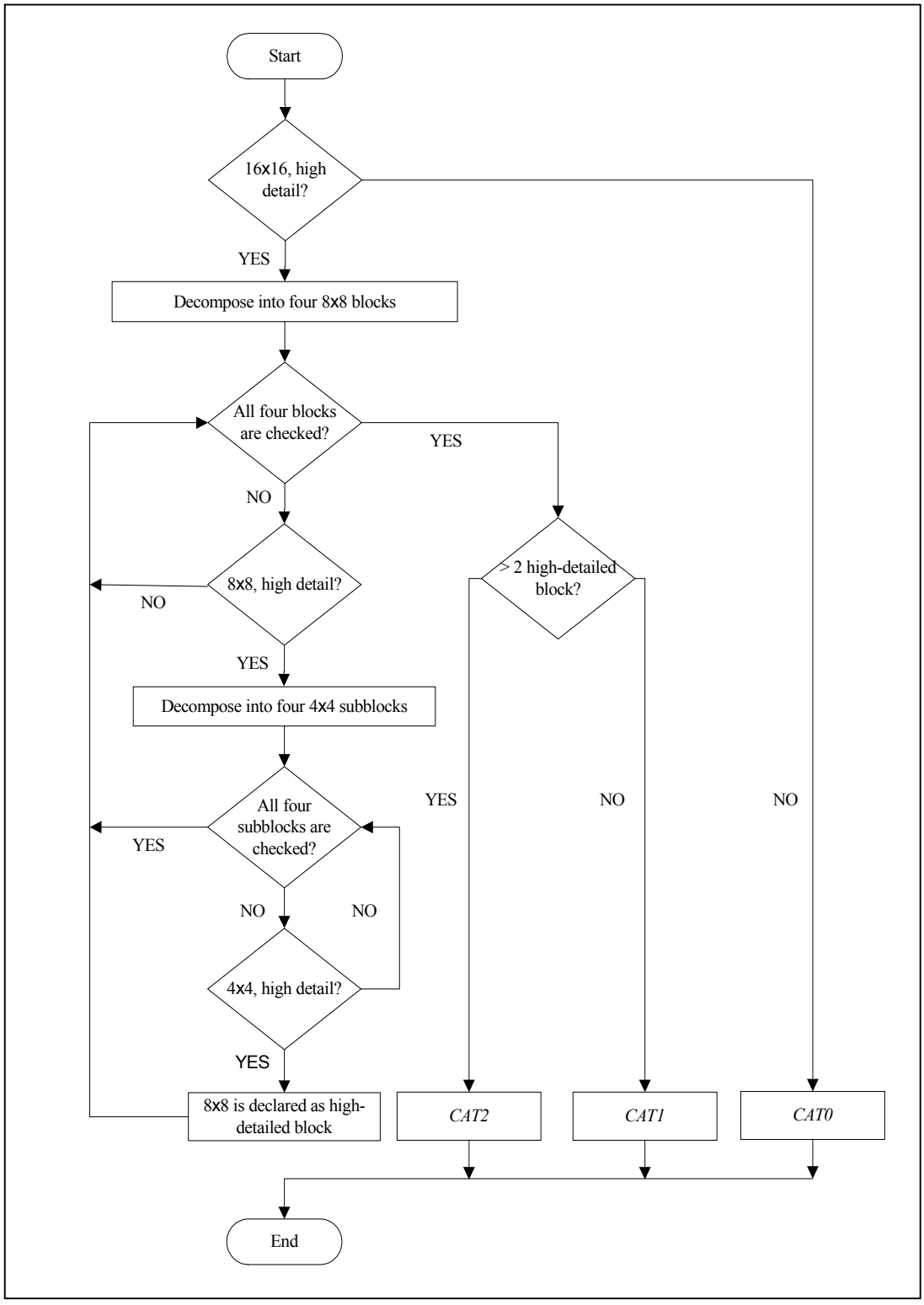

Fig. 3-2 The flowchart diagram of the proposed *Finter1* algorithm incorporates the complexity measurement for a macroblock.

final mode decision for the current macroblock.

Cat1 category algorithm:

C1. Obtain a motion vector for each of the four 8×8 blocks in a macroblock by using the full search algorithm with search range of ± 8 pixels.

C2. Continue to search for motion vector(s) for the 8×16 blocks, 16×8 blocks, and 16×16 macroblocks by referring only to the 4 search points, i.e., the motion vectors of the four 8×8 blocks.

C3. Perform step B2 to B3 to find the final mode decision for the current macroblock.

Cat2 category algorithm:

D1. Obtain a motion vector for each of the sixteen 4×4 blocks in a macroblock by using the full search algorithm with search range of ± 8 pixels.

D2. Continue to search for motion vector(s) for 8×4 blocks, 4×8 blocks, and 8×8 blocks by referring only to the 16 search points, i.e., the motion vectors of the sixteen 4×4 blocks.

D3. Perform the steps C2 to C3 to find the final mode decision for the current macroblock.

3.4 The proposed *Finter2* algorithm

The efficiency of the proposed *Finter2* is achieved by introducing two additional measurements targeted at two kinds of encoded macroblocks: (a) macroblocks encoded with *SKIP* mode (direct copy from the corresponding macroblock located at the same position in the previous frame); (b) macroblocks encoded by the inter-frame modes with larger decomposed partition size (greater than 8×8 pixels). By successfully identifying these two kinds of macroblocks, the encoder is exempted from examining them with all possible inter-frame modes, which saves encoding time.

Measurement of temporal similarity:

The *SKIP* mode is normally assigned to a macroblock that comprises almost identical pixel information to that of the corresponding macroblock in the same

position in the previous frame, for example in areas representing a static background. The macroblocks coded with *SKIP* mode (skipped macroblocks) can be easily detected by comparing the residue between the current macroblock and the previously encoded macroblock with a threshold as follows:

$$T(S_{\text{residue}}) = \begin{cases} 1, & S_{\text{residue}} < Th \\ 0, & S_{\text{residue}} > Th \end{cases} \tag{32}$$

$$S_{\text{residue}} = \sum_m \sum_n \left| \mathbf{B}_{m,n,t} - \mathbf{B}_{m,n,t-1} \right| \tag{33}$$

where S_{residue} is the sum absolute difference between $\mathbf{B}_{m,n,t}$ and $\mathbf{B}_{m,n,t-1}$, which represent current and previous macroblocks, respectively. If $T(S_{\text{residue}}) = 1$, the current macroblock is a skipped macroblock. However, performing this calculation for every macroblock further increases the encoding time. Lim et al. [10] suggested performing temporal similarity checking if the current 16×16 macroblock has zero motion. This necessitates each macroblock, including skipped macroblocks, to undergo at least one complete cycle of motion estimation. If the encoder can detect the skipped macroblocks without a priori knowledge, then a significant proportion of the encoding time will be saved.

Generally, the skipped macroblocks tend to occur in clusters, such as in a patch of static background. Thus, we propose that the current macroblock has to undergo temporal similarity detection if one of the encoded neighbours is a skipped macroblock. The temporal similarity detection is implemented according to (32) and (33), but we propose an adaptive spatially varying threshold, Th_{ASV}, to replace Th.

$$Th_{\text{ASV}} = C * \min\left(S_{N1}, S_{N2}, S_{N3}, S_{N4}\right) \tag{34}$$

where C is a constant; S_{N1}, S_{N2}, S_{N3}, and S_{N4} are the sum absolute difference of four nearest encoded neighbours, N_1, N_2, N_3, N_4, as shown in Fig. 3-3. They are valid and pre-stored in the system if and only if their corresponding macroblocks are skipped macroblocks. Thus, (34) reduces in size according to the number of skipped neighbouring macroblocks.

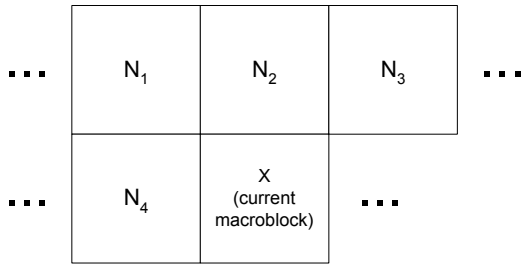

Fig. 3-3 The relative position of four nearest encoded neighbours of the current macroblock.

Measurement on block-based motion consistency:

The tendency that the inter-frame modes with larger partition size (of sizes 8×8, 8×16, 16×8, and 16×16 pixels) are more suitable to encode homogeneous macroblocks has been verified by a number of authors [7,10-11]. By contrast, macroblocks containing moving features appear more detailed and therefore require use of smaller block sizes. Thus, the proposed algorithm suggests checking the motion vector of each 8×8 block decomposed from a highly detailed macroblock. If consistency among motion vectors exists, the proposed algorithm checks the inter-modes with partition size greater than 8×8, otherwise, all possible inter-frame modes are searched.

The algorithm of *Finter2*:

Fig. 3-4 shows a flowchart of the proposed *Finter2* algorithm, which is summarized as follows:

E.1. Turn off all flags including *SKIP*, *INTRA* and all inter-modes

E.2. Check if one of the four nearest neighbours of the current macroblock is a skipped macroblock. Implement E.3 if the situation is true. If not, go to E4.

E.3. Obtain a threshold, Th_{ASV}, from (34). Compare Th_{ASV} with the sum absolute difference between the current macroblock and the previous macroblock at the same position. If the sum is smaller than the threshold, turn on the flag for *SKIP* only. Otherwise, continue to E.4.

E.4. Check the complexity of the macroblock using equation (29-a) and (31). If the current macroblock is homogeneous, turn on the flag for *I4MB*, *I16MB* and the inter-mode with partition size 16×16. Otherwise, continue to E.5.

E.5. Decompose the highly detailed macroblock into four non-overlapping 8×8 blocks. Check whether the motion vectors of the four blocks are consistent. If

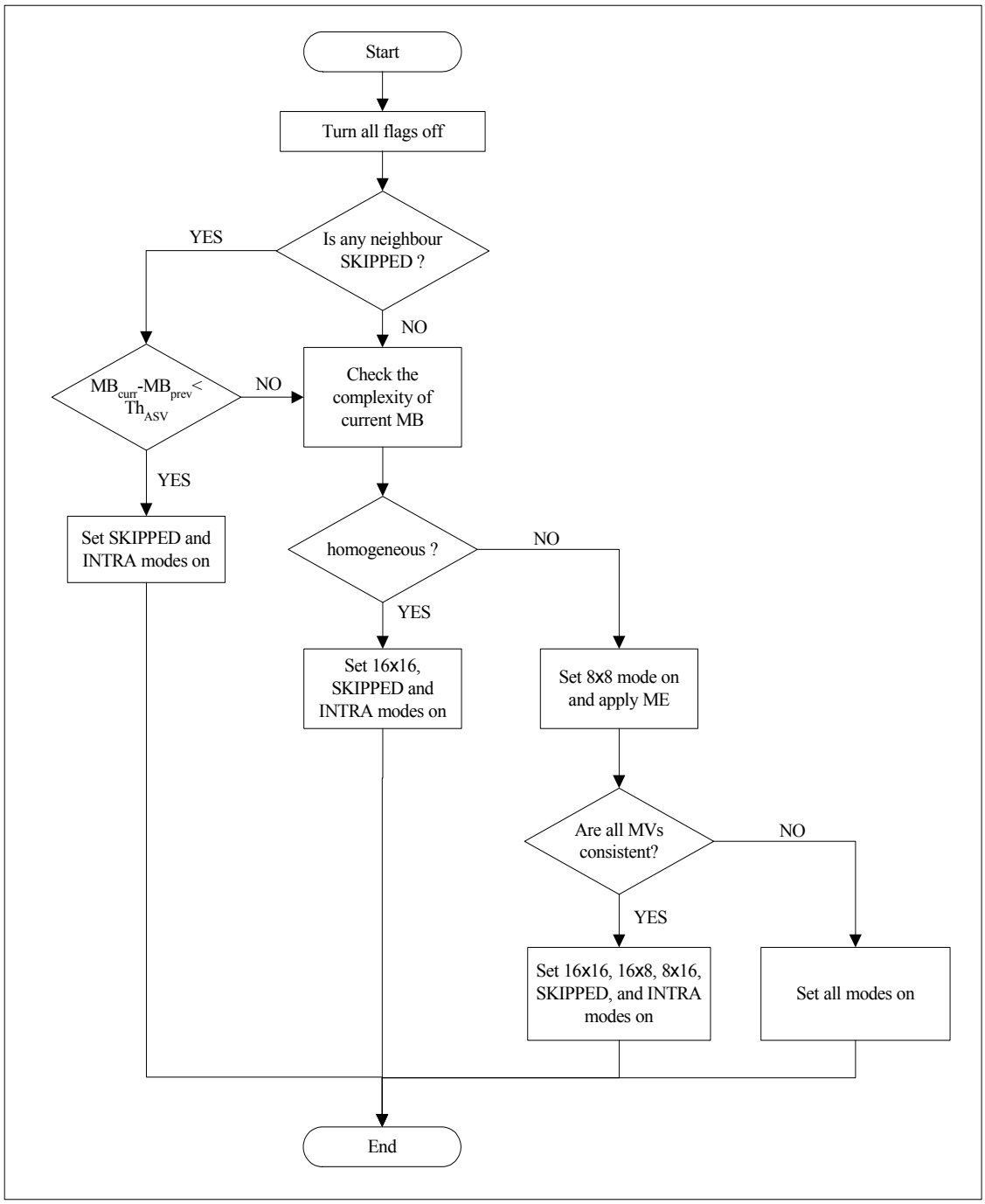

Fig. 3-4 Flowchart of the proposed *Finter2* algorithm incorporating the complexity measurement for a macroblock, temporal similarity, and the detection of different moving features within a macroblock.

consistent, check the flags of *I4MB*, *I16MB*, and the inter-modes with partition size 8×16, 16×8, and 16×16 and go to E.7. Otherwise, continue to E.6.

E.6. Turn on all flags and use sixteen motion vectors obtained from inter-modes with partition size 4×4 as searching points for the inter-mode with partition size 4×8 and 8×4 rather than performing full search. Then, continue to E.7.

E.7. Utilise the four motion vectors obtained from four 8×8 blocks as searching points for the inter-modes with partition size 8×16, 16×8, and 16×16 rather than performing full search.

3.5 Simulation results

This section compares two simulation results employing the proposed *Finter1* and *Finter2* algorithms. The settings of the simulations are as follows: all the sequences are defined in a static coding structure, i.e., one I-frame is followed by nine P-frames (1I9P), with a frame rate of 30 frames per second and no skip frame throughout the 100 frames. The precision and search range of the motion estimation is set to ¼ pixel and ±8 pixels, respectively. Lastly, Context-based Adaptive Binary Arithmetic Coding (CABAC) is used to perform entropy coding and a static quantizer value, Qp = 29, is applied throughout the simulation.

The table summarises the simulation results of the two algorithms – *Finter1* and *Finter2* in terms of PSNR difference, bit rate difference and speed up compared with JM6.1e, the testing benchmark. The general trends are identified as follows: both fast algorithms introduce less than 0.08 dB of PSNR degradation in Class A and Class B, and approximately 0.10 dB in Class C. Note that there is insignificant PSNR difference between the MFInterms and FInterms algorithms. As to compression ratio, the proposed MFInterms produces slightly higher bit rates than FInterms especially in the Class C sequences, however the bit differences for most test sequences are less than 5%. Nevertheless, the picture degradations and bit rate increase are generally considered within acceptable range as human visual perception is unable to distinguish a PSNR difference of less than 0.2dB.

Significantly, the new MFInterms algorithm provides a saving of 28-50% in encoding time for Class C sequences when compared with the JM6.1e benchmark. The saving for Class A and B sequences is 60-73%. The previously reported FInterms

algorithm provided improvements of only 18-25% and 22-31%, respectively. The reason that a significant proportion of the encoding time is saved with the MFInterms algorithm is that skipped macroblocks are detected accurately and are encoded with *SKIP* mode, obviating the need for other mode examinations. As a result, the encoding time of a P frame could be shorter than that of an I frame if a sequence contains a significant number of skipped macroblocks.

TABLE 3-2
Simulation results of the proposed *Finter1* and *Finter2* algorithms compared with JM6.1e, the H.264/AVC software, in three sequence classes.

	Sequences	PSNR Difference		Bit Rate Difference		Speed up cf. JM6.1e	
		Finter1	*Finter2*	*Finter1*	*Finter2*	*Finter1*	*Finter2*
A	Ship Container	-0.02 dB	-0.06 dB	0.10 %	0.37%	30.85%	72.94%
	Sean	-0.05 dB	-0.07 dB	0.44 %	-0.11%	29.87%	69.63%
B	Silent	-0.07 dB	-0.08 dB	1.47 %	3.73%	26.64%	60.22%
	News	-0.07 dB	-0.07 dB	1.34 %	1.52%	22.04%	62.08%
C	Stefan	-0.09 dB	-0.13 dB	5.36 %	8.90%	18.34%	28.72%
	Table Tennis	-0.08 dB	-0.09 dB	5.26 %	6.57%	24.74%	49.41%

Chapter 4
Comparison results of the combined algorithms

This section presents two sets of simulation results employing the proposed combinations of (*Fintra* + *Finter1*) algorithms and (*Fintra* + *Finter2*) algorithms for inter-frame coding, as P-frames may contain I-macroblocks. All the simulations were programmed using C++. The computer used for the simulations was a 2.8GHz Pentium 4 with 1024MB RAM. The testing benchmark was the JM6.1e version provided by the Joint Video Team (JVT) [12]. The selected sequences in 2 different resolutions, namely, QCIF (144×176) and CIF (288×352) formats, are classified into three different classes, i.e., Class A, which are sequences containing only low spatial correlation and motion, e.g., *Akiyo* and *Ship Container*; Class B, which contains medium spatial correction and/or motion, e.g., *Foreman* and *Silent Voice*; and Class C, where high spatial correlation and/or motion are involved, e.g., *Mobile & Calendar* and *Stefan*. In the following simulations, 22 test sequences in different resolutions are presented. Fig. 4-1 shows snapshot frames from the less common sequences used.

The test settings in inter-frame mode are as follows: all the sequences are defined in a static coding structure, i.e., one I-frame is followed by nine P-frames (1I9P), with a frame rate of 30 frames per second and no skip frame throughout the 300 frames. The precision and search range of the motion vectors are set to ¼ pixel and ± 8 pixels, respectively. *Fintra* is used for obtaining the best member from *I4MB*. Context-based Adaptive Binary Arithmetic Coding (CABAC) is used to perform the entropy coding and a static quantizer factor, *Qp*=32, is applied throughout the simulation. Since the mode decision of the two chrominance components (U and V) are affected when applying the proposed fast algorithms, *Finter1* and Finter2, the simulation results are

Fig. 4-1 Snapshot frames of the less common sequences used: (top left to right) City (Class B); Crew and Harbour (Class C); (bottom left to right) Paris (Class B); Template and Waterfall (Class C).

presented in terms of an average PSNR of the luminance and two chrominance components, i.e., Y, U, and V (measured in dB) rather than the PSNR of the luminance component (Y-PSNR).

Table 4-1 summarise the performance of two proposed combinations of algorithms, (*Fintra* + *Finter1*) and (*Fintra* + *Finter2*). The general trends are identified as follows: on average, there is a degradation of 0.02 dB in Class A, and approximately 0.05dB - 0.06 dB in other classes for both proposed combinations of algorithms. It is clear that the average PSNR difference between two proposed combinations of algorithms is insignificant (less than 0.02 dB). As to compression ratio, the tendency of a slight bit increase is directly proportional to the class of sequence. The test sequences of Class A attain the least bit increase, whereas the high motion sequences in Class B and Class C produce slightly higher bit rates than the H.264/AVC standard. However, the bit differences for most test sequences are still within an acceptable range of less than 5%. In general, the combination of (*Fintra* + *Finter2*) performs better in terms of compression than the combination of (*Fintra* + *Finter1*). The degradations and the bit differences are due to the erroneous prediction in the proposed combined algorithms. Nevertheless, the degradations are still below

the human visual threshold, which is widely recognised as being less than 0.15-0.20 dB.

TABLE 4-1
Simulation results of the two proposed combined algorithms, namely, *Fintra* + *Finter1* and *Fintra* + *Finter2*, versus the JM6.1e, H.264/AVC software, for three sequence classes and two resolutions.

Classes / Sequences		Resolution (pels)	Average PSNR Difference (dB)		Bit Rate Difference (%)		Speed-up cf. JM6.1e (%)	
			Fintra + Finter1	*Fintra + Finter2*	*Fintra + Finter1*	*Fintra + Finter2*	*Fintra + Finter1*	*Fintra + Finter2*
A	Akiyo	288×352	-0.01 dB	-0.02 dB	2.02%	0.75%	61.83%	81.64%
	Container Ship	144×176	-0.03 dB	-0.03 dB	1.60%	0.81%	60.35%	82.55%
	Grandma	144×176	-0.02 dB	-0.02 dB	1.26%	-0.02%	62.94%	79.86%
	Sean	144×176	-0.03 dB	-0.04 dB	2.11%	0.38%	60.19%	86.61%
B	City	288×352	-0.04 dB	-0.05 dB	4.70%	4.10%	57.06%	63.00%
	Coastguard	144×176	-0.03 dB	-0.05 dB	6.47%	7.26%	55.53%	59.53%
	Foreman	288×352	-0.06 dB	-0.08 dB	7.21%	6.53%	62.00%	66.00%
	News	144×176	-0.06 dB	-0.07 dB	3.17%	2.05%	60.36%	79.27%
	Paris	288×352	-0.03 dB	-0.05 dB	3.76%	3.74%	49.78%	71.66%
	Silent Voice	144×176	-0.04 dB	-0.05 dB	3.86%	4.75%	44.93%	67.35%
C	Car Phone	144×176	-0.09 dB	-0.10 dB	4.50%	3.50%	58.87%	62.17%
	Football	144×176	-0.09 dB	-0.11 dB	5.36%	6.78%	53.29%	50.54%
	Harbour	288×352	-0.05 dB	-0.04 dB	2.62%	0.59%	51.16%	52.39%
	Mobile & Calendar	144×176	-0.05 dB	-0.05 dB	2.33%	0.25%	51.53%	53.97%
	Stefan	288×352	-0.05 dB	-0.07 dB	2.01%	2.07%	52.17%	54.61%
	Table Tennis	144×176	-0.06 dB	-0.06 dB	6.87%	6.63%	58.98%	70.06%
	Template	288×352	-0.07 dB	-0.08 dB	2.29%	0.53%	54.65%	54.32%
	Waterfall	288×352	-0.02 dB	-0.02 dB	2.06%	0.79%	57.69%	68.00%

In contrast, the efficiency of the two proposed combined algorithms is far greater than that of JM6.1e, the benchmark: on average, a saving of 61% in encoding time for (*Fintra* + *Finter1*) and 83% for (*Fintra* + *Finter2*) for Class A. The speedup for other classes ranges between approximately 44-62% and 50-79% for two proposed combined algorithms, respectively. It is interesting to observe that the combination of (*Fintra* + *Finter2*) results in a significant speedup for Class A test sequences, while the performance drops (approximately 20%) when the spatial correlation/motion of the test sequences increase. The explanation is that the skipped macroblocks, which benefit the coding of the proposed *Finter2* algorithm, are abundant in low motion

sequences, while the high motion sequences require more predictions from previous frames rather than direct copy. Thus, the advantage of *Finter2* becomes less significant in Class B and Class C test sequences. In any case, the general speedup of both proposed combined algorithms is in excess of 50%.

Chapter 5
Conclusions and future prospects

5.1 Main contributions

In this report, we proposed three fast algorithms, namely, *Fintra*, *Finter1*, and *Finter2*, for fast mode selection in the H.264/AVC standard. The algorithms improve the computational performance of both intra- and inter-frame coding, thus reducing the implementation requirements in real applications. Improvement is accomplished by discarding the least possible modes to be selected. The *Fintra* algorithm intelligently selects fewer candidate members need to undergo expensive Lagrangian evaluation. The *Finter* algorithms utilise a complexity measure to identify those low detailed macroblocks that require less demanding processing. The results of extensive simulations demonstrate that the proposed algorithms can attain a time saving of up to 62% and 86% in intra-coding and inter-coding, respectively, compared with JM6.1e, the benchmark. This is achieved without sacrificing both picture quality and bit rate efficiency.

5.2 Timetable of the previous research projects

I joined the department of Computer Science at the University of Warwick on 1st October 2003. It has been nine months until the end of June 2004.

My research interest is generally extended from the previous achievements obtained at Nanyang Technological University in Singapore where I pursued research studies before transferring to United Kingdom. I have submitted five papers (including one journal paper and four conference papers) in October 2003, December 2003, March 2004, and May 2004, respectively. The detailed timetable (table 5-1) presented in the following will give a clear summary:

TABLE 5-1
The table specifies the important events and the dates October 2003 to June 2004.

Date/Period	Descriptions
01st Oct. 2003	Joined Department of Computer Science at Warwick University
24th Oct. 2003	Submitted a paper to *IEEE International Conference on Acoustics, Speech and Signal Processing* (*ICASSP*) *2004* in Montreal, Canada.
04th Nov. 2003	Conducted a group seminar for multimedia group in Department of Computer Science.
15th Dec. 2003	Submitted a paper to *IEEE International Conference on Multimedia and Expo* (*ICME*) *2004* in Taipei, Taiwan.
09th Jan. 2004	A paper was accepted by *ICASSP 2004*.
01st Mar. 2004	A paper was accepted by *ICME 2004*.
Dec. 2003 – Mar 2004	Worked on progressive fast algorithm for inter-frames in H.264/AVC.
15th Mar. 2004	Submitted two papers to *IEEE International Conference on Image Processing* (*ICIP*) *2004* in Singapore.
01st May 2004	Submitted one paper to Special Issue on Emerging H.264/AVC video coding of *Journal of Visual Communication and Image Representation*.
20th May 2004	A paper was accepted by *ICIP 2004*. Another was rejected.
30th Jun. 2004	A presentation was given on Postgraduate Research Day.
May 2004 – June 2004	A new project, scalable video coding, is started.

5.3 Future prospects

My future research interests will focus on the emerging video coding standard, Scalable Video Coding (SVC), the most recent standard, which is championed by the Joint Video Team (JVT), a collaboration team of two international standard bodies, ITU-T VCEG and ISO/IEC MPEG.

The procedure of ASVC standard

The JVT called for proposal on scalable video coding technology in October 2003 at the meeting convened in Brisbane, Australia [17]. The aim was to select the most promising proposals as the starting point of a prospective Advanced Scalable Video Coding (ASVC) standard with high compression performance. The JVT plan to develop the ASVC standard within a time frame of approximately next 2-3 years. Until the submission deadline in December 2003, 26 pre-registrations had been received. But only a total of 21 submissions were finally made [15]. In early March, the JVT declared the algorithms proposed by four institutes and companies following selection at the meeting in Munich, Germany. The winners are Heinrich Hertz

Institute (HHI) of Germany, Microsoft Research Asia (MSRA), National Chiao-Tung University (NCTU) of Taiwan, and Poznań University of Technology (PU) of Poland [16]. In late March, the JVT called for another proposal on core experiments (CE) of ASVC standard. The CEs aim at obtaining significant improvements in four parts of the ASVC standard: scalable motion information (CE1a), spatial transform and entropy coding (CE1b), intra modes (CE1c), base-layer of scalable video (CE1d). The details of each CE are elaborated in Table 5-2.

Since core experiments CE1a and CE1c have some correlation with my previous research interest, my research will concentrate on these two core experiments. The two major research works include: (1) studying and learning how to utilise the codec of the CE provided by MPEG SVC group, (2) improving the performance of the state-of-the-art scalable video techniques.

TABLE 5-2
The table describes the Core Experiments of MPEG SVC proposed by JVT.

Core Experiments (CE)		Descriptions
1	a. Scalable motion information	Evaluating the optimum trade-off between motion data and texture data depending on the spatial-temporal-SNR resolution.
	b. Spatial transform and entropy coding	Evaluating different spatial transforms and associated entropy coding techniques.
	c. Intra modes	Improving coding efficiency using intra coding.
	d. Introduction of a base-layer	Improving the performance of the scalable coder at low resolutions (low image size and frame rate).
	e. MCTF with deblocking	Improving visual quality and coding efficiency by reducing artefacts occurring in t+2D wavelet video coding schemes with block-based motion compensation.
2	AVC-based CE	NIL
3	Quality evaluation	Ensuring that the above visual quality evaluation method for the CE is valid.

The structure of the scalable video coding

Scalable video coding technique enables an encoder to arrange the coded stream in a number of layers, including a base layer and several enhancement layers (Fig 5-1). The decoder can optimise the video quality over a given bit rate range by selecting part of the coded bitstream. A basic quality sequence is obtained if a decoder selects to receive the base layer bitstream, whereas, a higher quality sequence is presented if more enhancement layers are received as well as the base layer.

The ASVC standard supports a number of scalable coding modes: spatial scalability, temporal scalability, Signal-to-noise (SNR) scalability, and fine-granularity scalability (FGS). Spatial scalability enables a video sequence coded at a hierarchy of spatial resolutions. Temporal scalability codes a video sequence into two layers at the same spatial resolution but different frame rates. SNR scalability supports different quantisation accuracy in different layers at the same frame rate and spatial resolution. Finally, FGS enables the quality of the sequence to be increased in small steps [22].

The scalable video coding technique defined in the ASVC standard is different from other standards in that it makes use of Overcomplete Discrete Wavelet Transform (ODWT) rather than a traditional block-based transform. In addition, ASVC supports Motion-Compensated temporal filtering (MCTF), performs wavelet subband decomposition along the temporal axis, while concentrating most of the energy of the GOP towards the low-frequency subband [23]. The supporting wavelet includes Haar MCTF, 5/3 MCTF [14].

Application of the scalable video coding

A set of applications require the support of scalable and reliable video coding, for instance, wireless systems with variable and fading bandwidth, video broadcasting in heterogeneous communication networks, mobile and wireless LAN video for multi-party conversational services, etc [19-21].

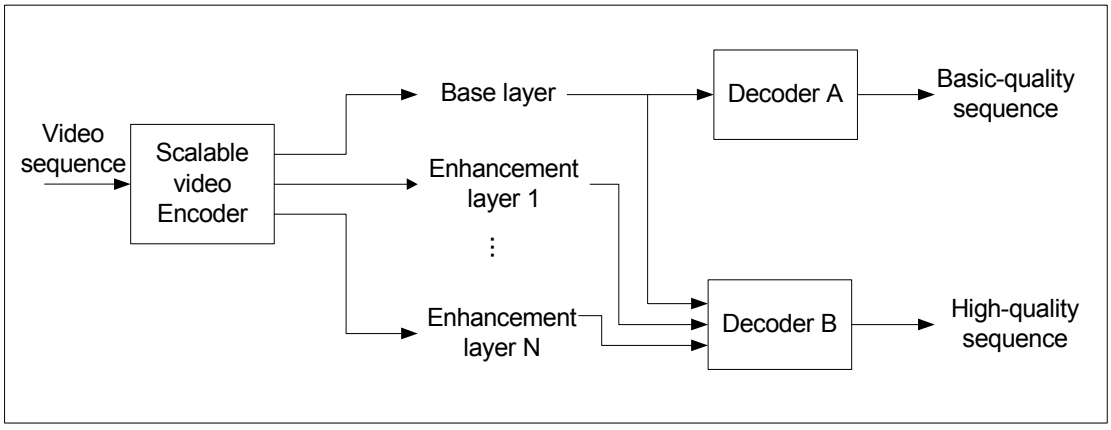

Fig 5-1 General concept of the scalable video coding technique.

Application Example 1: video streaming over heterogeneous IP networks

Due to rapid development of network transmission capacity, increasing numbers of on-demand video contents have to be streamed to the end user either through wired or wireless connection. There exist many bottlenecks in the current non-scalable streaming video technologies:

- The connection between server and clients vary from one user to another:
  Wireless: GSM, GPRS, 3G, wireless LAN, Bluetooth.
  Wired: dial-up, ISDN, cable, ADSL, LAN, WAN.
- The connection bandwidth ranges between 9.6Kb/s to 100Mb/s.
- The user device capability is quite different: PDA, laptop, set-top box, TV.
- Different memory, computational power, and screen resolution.

Scalable video coding technology featuring fine-granularity provides a solution for the above issues, for instance, the bit rate of streaming video can be adjusted to adapt to the channel bandwidth and to reduce the channel congestion. The layer structure feature in scalable video coding can apply unequal protection (UEP) to more important layers without causing catastrophic distortion in visual quality when the bandwidth of transmission becomes narrow. Multicast servers just need to transmit the base layer to all users in order to save bandwidth. Different users can subscribe to the bitstreams with different bit-rates as they are required.

Application example 2: storage applications

The distribution of video content over a variety of network connections to different storage devices is in great demand, for example, CCTV video through a surveillance network demands a low storage requirement. Scalable video coding technology has the advantage of providing “selective degradation”, i.e., the quality can be adjusted based on the available storage capacity.

5.4 List of publications

Journal paper:

[1] Andy C. Yu, Ngan King Ngi, and Graham Martin, "Efficient Intra- and Inter-mode Selection Algorithms for H.264/AVC", accepted by special issue of *Journal of Visual Communication and Image Representation*.

Conference papers:

[2] Andy C. Yu, "Efficient Prediction Mode Selection Algorithm for Intra frame Coding in H.264/AVC", will submit to *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 2005*, Philadelphia, March 2005.

[3] Andy C. Yu and Graham Martin, "Advanced Block Size Selection Algorithm for INTER frame Coding in H.264/AVC" to appear in *Proc. of IEEE International Conference on Image Processing (ICIP) 2004,* Singapore, Oct 2004.

[4] Andy C. Yu, Bing Zeng, Oscar Au, "Arbitrarily Shaped Video Coding: Smart Padding versus MPEG-4 LPE/Zero Padding", to appear in *Proc. of IEEE International Conference on Multimedia & Expo (ICME) 2004*, Taipei, Taiwan, June 2004.

[5] Andy C. Yu, "Efficient Block Size Selection Algorithm for INTER frame Coding in H.264/AVC", to appear in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 2004*, Montreal, Canada, May 2004.

[6] Andy C. Yu, Oscar Au, Bing Zeng, "Removing of Blocking Artefacts Using Error-Compensation Interpolation and Fast Adaptive Spatial-Varying Filtering", *Proc. of IEEE International Symposium on Circuit and System (ISCAS) 2002*, vol. 5, pp. 241-244, Scottsdale, USA, May 2002.

[7] Andy C. Yu, Bing Zeng, Oscar Au, "A Novel Motion Estimation Algorithm for arbitrarily shaped video coding", *Proc. of IEEE International Conference on Multimedia & Expo (ICME) 2002, vol. 1, pp. 649-652,* Lausanne, Switzerland, August 2002.

Dissertation:

[8] Andy C. Yu, "Arbitrarily-shaped Video Coding", published by Department of Electronic, Electrical Engineering of Hong Kong University of Science and Technology (HKUST), Hong Kong.

Technical report:

[9] Andy C. Yu, "Multi Reference Frame Structure Feature with Long-term reference frames", full proposal for STMicroelectronics Asia Pacific Pte. Ltd.

References

[1] "Information technology - coding of audio visual objects - Part 10: advanced video coding," ISO/IEC 14496-10:2003, Dec. 2003.

[2] Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, "Joint Model Number 1 (JM-1)," JVT-A003, Jan. 2002.

[3] T. Wiegand and B. Girod, *Multi-frame motion-compensated prediction for video transmission*, published by Kluwer Academic Publishers, 2001.

[4] H. Everett III, "Generalize Lagrange multiplier methods for solving problems of optimum allocation of resources," *Operations Research*, vol. 11, pp. 399-417, 1963.

[5] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 36, no. 9, pp. 445-1453, Sep. 1988.

[6] P. Chou, T. Lookabaugh, and R. Gray, "Entropy-constrained vector quantization," *IEEE Trans. on Acoustics, Speech And Signal Processing*, vol. 37, no. 1, pp. 31-42, Jan. 1989.

[7] I. Richardson, *H.264 and MPEG-4 video compression*, published by Wiley, 2003

[8] M. Kankanhalli, Rajmohan, and Ramakrishnan, "Content-based watermarking of image," http://www.acm.org/sigs/sigmm/MM98/electronic_proceedings/toc.html.

[9] JVT reference software, JM 6.1e, http://bs.hhi.de/~suehring/tml.

[10] K. Lim, S. Wu, J. Wu et al., "Fast inter mode selection," JVT-I020, ISO/IEC JTC1/SC29/WG11 and ITU-I SG16 Q.6, Sep. 2003, San Diego, U.S.

[11] A. Yu, "Efficient block-size selection algorithm for inter-frame coding in H.264/MPEG-4 AVC," to appear in *proceedings of international conference on Acoustics, Speech and Signal Processing (ICASSP) 2004*, May 2004, Montreal, Canada.

[12] JVT reference software, JM6.1e, downloaded from http://bs.hhi.de/~suehring/tml.

[13] T. Wiegand, G. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE trans. on Circuit and Systems for Video Technology*, vol. 13, no. 7, pp. 560 –576, 2003.

[14] J. Reichel, K. Hanke, B. Pesquet-Popescu, "Scalable video model v. 1.0,"N6372, ISO/IEC JTC1/SC29/WG11, March 2004, Munich, Germany.

[15] __, "Registered responses to the call of proposal on scalable video coding," M10569, ISO/IEC JTC1/SC29/WG11, March 2004, Munich, Germany.

[16] M. van der Schaar and M. Domański, "Description of core experiments in SVC," N6373, ISO/IEC JTC1/SC29/WG11, March 2004, Munich, Germany.

[17] __, "Call for proposals on scalable video coding technology," N6193, ISO/IEC JTC1/SC29/WG11, Dec 2003, Waikoloa, USA.

[18] __, "Requirements and applications for scalable video coding," N6025, ISO/IEC JTC1/SC29/WG11, Oct 2003, Gold Coast, Australia.

[19] Ł. Błaszak, M. Domański, R. Lange, and A. Łuczak, "Scalable AVC codec," M10626, ISO/IEC JTC1/SC29/WG11, March 2004, Munich, Germany.

[20] M. van der Schaar, C. Tsai, and T. Ebrahim, "Report of ad hoc group on scalable video coding," M9076, ISO/IEC JTC1/SC29/WG11, Dec 2002, Awaji Island, Japan.

[21] D. Wu, Y. Hou, Y. Zhang, "Scalable video coding and transport over broadband wireless networks," *Proceedings of the IEEE*, vol. 89, pp. 6-20, January 2001.

[22] W. Li, "Overview of fine granularity scalability in MPEG-4 video standard," *IEEE trans on Circuit and Systems for Video Technology*, vol. 11, no. 3, Mar. 2001.
[23] S. Tubaro and G. Cordara, "Fast in-band MCTF with scalable motion vectors," N10569-S01, ISO/IEC JTC1/SC29/WG11, Mar 2004, Munich, Germany.
[24] A. Luthra, G. Sullivan, T. Wiegand, "Introduction to the special issue on H.264/AVC video coding standard," *IEEE trans. on Circuit and Systems for Video Technology*, vol. 13, no. 7, pp. 557 –599, 2003.
[25] A. Netravali and B. Haskell, "Digital pictures, representation and compression and standards," second edition, Plenum Press, New York, 1995.
[26] T. Tan, M. Ghanbari, and D. Pearson, "An objective measurement tool for MPEG video quality," *Signal Processing*, vol. 7, pp. 279-294, 1998