Motion compensation and bit-rate control techniques for video encoder optimization

Jing, Xuan

2009

Jing, X. (2009). Motion compensation and bit-rate control techniques for video encoder optimization. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/15560>

<https://doi.org/10.32657/10356/15560>

MOTION COMPENSATION AND BIT-RATE CONTROL TECHNIQUES FOR VIDEO ENCODER OPTIMIZATION

JING XUAN

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement for the degree of
Doctor of Philosophy

2009

Acknowledgements

First of all, I would like to thank my supervisor, Prof. Lap-Pui Chau, for his invaluable guidance, constant encouragement and support throughout the whole process of my graduate study at Nanyang Technological University. I am also very grateful to Prof. Yap-Peng Tan for his constructive feedback and suggestion during my study at NTU.

I take this opportunity to thank Prof. Ce Zhu for his guidance and assistance to my graduate research work.

I would like to thank my colleagues in Media Technology Lab and Information Systems Research Lab, Jinghong Zheng, Yunlong Gao, Haiyan Shu, Tao Fang, Yu Wang, Boon Seng Chew and other members of the lab for their friendship and help. It was a great pleasure to work and study with all of them. They have made my life and work at NTU an enjoyable experience.

Last but not least, I would like to thank my family for their love and endless support. They have taught me how to deal with problems and how to adapt myself to the challenges of life. Without their support, this work would never have been accomplished.

Contents

Acknowledgements	i
Contents	ii
Summary	v
List of Figures	vii
List of Tables	xi

1 Introduction.....	1
1.1 Motivation	1
1.2 Objective and Main Contributions	4
1.3 Outline of the Thesis	6
2 Overview of Motion Compensation and Rate Control Techniques.....	8
2.1 Typical Video Compression System	8
2.2 Block Matching Motion Compensation Techniques.....	11
2.2.1 Block Matching Method.....	13
2.2.2 Fast Search Strategy for Motion Estimation	15
2.2.3 Block Matching Criteria	20
2.2.4 Fast Matching Strategy for Motion Estimation	23
2.2.5 Variable Size Block Matching Motion Compensation.....	24
2.3 Rate Control for Video Coding	27
2.3.1 Quantization Process	27
2.3.2 Variable Quantization Scale Rate Control	29
2.3.3 Standard Rate Control Algorithms	32

2.4 Summary.....	35
3 Fast Block Matching Motion Estimation Algorithms	36
3.1 Introduction	36
3.2 Efficient Three-Step Search Algorithm.....	38
3.2.1 Observations	38
3.2.2 Review of New Three-Step Search Algorithm.....	40
3.2.3 Development of Efficient Three-Step Search Algorithm.....	41
3.2.4 Simulation Results.....	44
3.3 Partial Distortion Search Using Predictive Search Area	48
3.3.1 Traditional Adjustable Partial Distortion Search	48
3.3.2 Predictive Search Area Based Partial Distortion Search.....	50
3.3.3 Simulation Results.....	55
3.4 Remarks on Fast Search and Fast Matching.....	61
3.5 Conclusion	62
4 Smoothness Constrained Block Matching Criterion.....	64
4.1 Introduction	64
4.2 Smooth Constrained MAD Criterion.....	65
4.2.1 Motivation	66
4.2.2 Development of the SCMAD	66
4.3 Experimental Results.....	69
4.4 Conclusion	75
5 Efficient Inter Mode Decision for H.264/AVC Video Coding	76
5.1 Introduction	76
5.2 Standard Mode Selection Scheme in H.264/AVC	77
5.3 Efficient Inter Mode Decision Algorithm	80
5.3.1 Macroblock Classification.....	80
5.3.2 The Algorithm for Inter Mode Decision	82
5.4 Experimental Results and Discussions.....	84
5.5 Conclusion.....	89
6 Rate Control for H.264/AVC with Frame Complexity Estimation	90
6.1 Introduction	90
6.2 Background and Related Work	92

6.2.1 Characteristics of H.264/AVC Rate Control	92
6.2.2 Related Work for Rate Control.....	94
6.3 Intra-frame R-Q Modeling	95
6.3.1 Traditional Method in JVT	95
6.3.2 Gradient-based Complexity Measure for Intra-frames.....	96
6.3.3 Adaptive Intra-frame R-Q Model	99
6.3.4 Model Performance Evaluation	102
6.4 Frame MAD Prediction and Frame Level Bit Allocation	104
6.4.1 R-Q Model and Linear MAD Prediction Model	104
6.4.2 Histogram Based Frame MAD Prediction Model	106
6.4.3 MAD Prediction Model Performance Evaluation	109
6.4.4 Improved Frame Level Bit Allocation Scheme.....	111
6.5 Experimental Results.....	116
6.6 Conclusion.....	119
7 Conclusions and Extensions.....	121
7.1 Summary of the Research.....	121
7.2 Future Extensions	123
Publications	126
Bibliography	128

Summary

Video compression has numerous applications in modern multimedia communications and storage such as digital television (DTV), Internet video streaming and digital versatile disk (DVD) video storage. The strong correlations between successive video frames and within individual frames make it possible to achieve high compression by reducing the temporal and spatial redundancies. Without video compression, all the digital video transmission and storage would not be practical. Various video compression standards (MPEG-1/2/4, H.261/263 and H.264/AVC etc.) have been developed since early 90's and they have been widely used throughout the years. Although these standards have specified the decompression procedure, many issues regarding the implementations of compression algorithms are open for optimization. In this thesis, we study various fundamental techniques in motion compensated video compression including motion estimation, mode decision and encoder rate control. The goal of our research is to achieve better tradeoff between encoding complexity and quality as well as better utilization of the available channel bandwidth.

We first investigate block matching based motion estimation techniques. In video coding, motion estimation is the most computation intensive task and therefore to achieve significant complexity reduction in motion estimation without reducing too much the estimation quality is highly desirable. We propose two fast algorithms based on the adaptive search pattern and partial distortion metric respectively. Both algorithms explore the statistical characteristics of motion vector distribution and can adaptively select suitable search points or block matching criteria for different areas within the search

window. Experimental results show that, by using the proposed fast approaches, the motion estimation complexity is greatly reduced with only slight degradation in estimation accuracy. Moreover, we propose a new measure of similarity between blocks in motion estimation called the smooth constrained mean absolute difference (SCMAD). Compared with the traditional MAD criterion, we achieve reduced bit rate for encoding the residue block without any degradation of the reconstructed image quality.

Variable size block motion estimation is a very important technique for video coding. The latest H.264/AVC standard employs seven different size block types in motion estimation which can significantly improve the coding performance compared with the previous video coding standards. In order to substantially reduce the computational complexity of the motion estimation and mode decision process in H.264/AVC, we introduce an efficient inter mode decision algorithm based on the macroblock (MB) classification. The proposed technique can save the computation cost significantly while maintaining the same rate-distortion (R-D) performance.

The main objective of rate control is to regulate the output bit rates of video encoders to achieve the best utilization of the available channel bandwidth. We propose two frame complexity based schemes for H.264/AVC video rate control. The first scheme is an adaptive intra-frame rate-quantization (R-Q) model using gradient-based frame complexity measure. This model aims at selecting accurate quantization parameters (QP) for intra-coded frames. In addition, we develop a frame level target bits allocation scheme which takes into consideration both the buffer status and the predicted frame MAD value. We show that the overall performance of the rate control, in terms of target bits mismatch and the visual quality of reconstructed video, can be improved.

List of Figures

1.1	A generic video communication system	2
2.1	Block diagram of a generic hybrid motion compensated video encoder	9
2.2	An example of frame configuration in a group of pictures (GOP)	10
2.3	Motion compensated frame and the corresponding residue frame. (a) Original current frame to be coded. (b) Difference between the current frame and its previous reference frame. (c) Motion compensated frame for frame (a) with corresponding motion vectors. (d) Prediction residue between the original frame and the motion compensated frame.....	12
2.4	Block matching motion estimation	14
2.5	Three-step search: a search path for finding MV (5,-7).....	16
2.6	Four-step search: two search paths for finding MV (7,-6) and MV (-7,3).....	17
2.7	Diamond search: a search path for finding MV (5,-2)	18
2.8	Hexagon-based search: a search path for finding MV (5,-4)	19
2.9	Performance comparison of different block matching criteria when coding <i>Foreman</i> sequence. The frame Mean Squared Error for different criteria is as follows: MSE 36.13, MAD 37.42, NCF 41.59, PDC (th=20) 40.36, PDC (th=30) 42.71, PDC (th=40) 49.49, MiniMax 41.57	22
2.10	Typical covering quadtree for VSBM [57]	25
2.11	Block structures for FSBM and VSBM for <i>Stefan</i>	26
2.12	Relation between QP and Qstep.....	28

2.13 VBR and CBR coding schemes with corresponding relations between the output bit rate and quality	30
2.14 Procedures for a typical video rate control scheme.....	31
3.1 Motion vector distributions of (a) <i>Football</i> and (b) <i>Foreman</i>	38
3.2 Multiple local minima within a search window	39
3.3 Two different search paths for finding MV within 5×5 area in N3SS	40
3.4 Search pattern used in the first step of E3SS	42
3.5 Block diagram of the E3SS algorithm.....	43
3.6 Search path for finding MV (3,-1) in E3SS	44
3.7 Frame by frame comparisons for <i>Football</i> sequence (720×486 , 59 frames) with search window $w = 15$. (a) MSE per frame. (b) Average search points per MB.....	47
3.8 Pixel grouping pattern for calculation of the partial distortions.....	49
3.9 Distribution of differential MVs	51
3.10 Locations of neighboring MBs.....	52
3.11 Formation of the predictive search area (PSA) around the predicted MV.....	53
3.12 Flowchart of the proposed partial distortion search with predictive search area	54
3.13 PSNR performance of the proposed partial distortion search algorithm against the quality factor k using sequence <i>Coastguard</i>	56
3.14 Frame by frame comparisons of average operations per MB for sequences (a) <i>Mobile Calendar</i> and (b) <i>Coastguard</i>	59
3.15 Frame by frame comparisons of probabilities of finding true MVs for sequences (a) <i>Mobile Calendar</i> and (b) <i>Coastguard</i>	60
4.1 Four 8×8 residue blocks used in the SCMAD.....	66
4.2 Block diagram of the proposed SCMAD	67

4.3	R-D performance of SCMAD with different α values. (a) PSNR difference vs. α and (b) Bitrate difference vs. α	70
4.4	Performance comparisons for coding the 67th frame of <i>Foreman</i> sequence. (a) Original frame (b) Decoded frame by using MAD as block matching criterion. The PSNR is 30.44 and the number of bits is 8258. (c) Decoded frame by using SCMAD as block matching criterion. The PSNR is 30.56 and the number of bits is 7493	72
4.5	Frame by frame comparisons for <i>Foreman</i> sequence (a) PSNR (b) Bit rate	73
4.6	Frame by frame comparisons for <i>Mother & Daughter</i> sequence (a) PSNR (b) Bit rate	74
5.1	Different block types used in H.264/AVC inter frame coding	78
5.2	R-D performance comparisons between VSBM and FSBM using H.264/AVC codec.....	79
5.3	Relations between frame differences and the block types used in H.264/AVC (a) Original frame from the sequence <i>Mother & Daughter</i> (b) Difference frame between the original frame and the previous frame (c) Decoded frame with corresponding macroblock types by the H.264/AVC encoder.....	81
5.4	Flowchart of the proposed inter mode decision algorithm.....	83
5.5	R-D performances of both algorithms for (a) QCIF sequences and (b) CIF sequences.....	87
5.6	Examples of the MB classification results and the final mode decisions for sequences <i>Foreman</i> and <i>Mother & Daughter</i> (a) Original frame (b) MB classification results. The MBs in white are classified as non-homogeneous. (c) Decoded frame with corresponding MB modes by the reference encoder with full mode search scheme. (d) Decoded frame with corresponding MB modes by the proposed method	88
6.1	<i>Chicken and egg</i> dilemma in H.264/AVC rate control	92
6.2	Intra bit rate of <i>Foreman</i> sequence under different QPs.....	96
6.3	Gradient per pixel and average pixel value for different frames of <i>Foreman</i>	97
6.4	Scatter plots of bit rate versus gradient value for <i>Foreman</i> sequence. Corresponding QP values (from top to bottom, left to right) are: 24, 28, 32 and 36	98

6.5	Curve fitting result for the first frame of <i>Suzie</i> . The gradient value $G = 9.74$ and the model parameters in $f(Q_{step}) = a \cdot Q_{step}^b$ are $a = 0.638$, $b = -0.795$	100
6.6	Intra-frame bit rate of different rate control schemes for <i>Comb1</i> sequence.....	103
6.7	Intra-frame QP determined by different rate control schemes for <i>Comb1</i> sequence	104
6.8	Abrupt scene change detection using <i>FD</i>	108
6.9	MAD prediction error comparisons for <i>News</i>	110
6.10	MAD prediction error comparisons for <i>Trevor</i>	111
6.11	Block diagram of the rate control framework for H.264/AVC	114
6.12	Frame by frame comparisons for <i>Trevor</i> sequence. (96kbps, 30fps, QCIF) (a) Luminance PSNR (b) Buffer fullness (c) Frame bits mismatch.....	118

List of Tables

3.1	Performance comparisons for CIF sequence <i>Coastguard</i>	45
3.2	Performance comparisons for CCIR601 sequence <i>Football</i>	45
3.3	Performance comparisons for CIF sequence <i>Suzie</i>	46
3.4	Performance comparisons for CIF sequence <i>Mother & Daughter</i>	46
3.5	Performance comparisons for CIF sequence <i>Salesman</i>	46
3.6	Comparisons for PSNR and probability of finding true MVs.....	57
3.7	Comparisons for average operations per MB and the speed-up gain against FS	57
3.8	MV predictor performance comparison for <i>Foreman</i> (D=3).....	58
3.9	Average PSNR degradations and speed-up ratios of different algorithms for all sequences.....	58
4.1	PSNR and bit rate comparisons for different block matching criteria	71
5.1	PSNR and bit rate comparisons between the full mode search and the proposed inter mode decision algorithm	85
5.2	Computational complexity comparisons between the proposed inter mode decision algorithm and the full mode search scheme	85
5.3	Performance comparison of different mode decision algorithms	86
6.1	Comparison of intra rate estimation performance.....	102

6.2	Correlation coefficients between actual frame MAD and the frame distance FD for different sequences	107
6.3	Comparison of MAD prediction accuracy	109
6.4	Comparisons of actual bitrate, frame skipping, average luminance PSNR and encoding time between JM (JVT-G012) rate control and our proposed rate control method.....	116
6.5	Comparisons of PSNR variation and frame bits mismatch between JM (JVT- G012) rate control and our proposed rate control method	117

Chapter 1

Introduction

1.1 Motivation

The conventional analogue video, due to its high bandwidth requirement for transmission and the difficulty in storage and manipulation, is no longer suitable for today's video communication applications. Compared with analogue video format, digital video can be more easily manipulated. Video compression technique makes it possible to transmit digital video content using much less bandwidth than the analogue video. Moreover, this low bandwidth requirement also enables efficient storage of digital video on the optical storage media or hard disk drives.

With the rapid growth in communication and video coding technologies, digital video applications have become more feasible than ever. Those applications include video conferencing, video telephony, digital television (DTV) and high definition television (HDTV) [7], digital video broadcasting (DVB) [6, 8] as well as entertainment applications such as digital versatile disk (DVD). Fig. 1.1 shows the block diagram of a generic

Figure 1.1. A generic video communication system

video communication system. The strong correlations among the input source video frames make it possible to achieve high compression by reducing the temporal and spatial redundancies. This compression is performed by the most essential part of this whole system: the video encoder. An efficient video encoder can choose appropriate coding algorithms according to the computing power constraint of the system. In addition, it can adjust its encoding parameters to regulate the output bit rate thus make the maximum use of the available transmission channel bandwidth and avoid the undesirable buffer overflow and underflow. The compressed bitstream will be transmitted through the communication channel and then it will be received by the receiver. Finally, the video decoder decompresses the bitstream from the decoder buffer and the reconstructed video frames are displayed at the user end.

Since early 90's, various video coding standards have been developed by the International Telecommunication Union-Telecommunication Standardization Sector (ITU-T) and the International Standards Organization (ISO) such as H.261 [1], H.262/MPEG-2 [4], H.263 [2], MPEG-1 [3], MPEG-4 [5] and the latest H.264/AVC [76]. Although these standards have specified the bitstream syntax and the decompression process, many issues regarding the implementations of compression algorithms for video encoder are open for optimization. Taking the most widely used motion compensated video encoder as an

example. Compared with the corresponding decoding process, the major difference between the encoder and the decoder lies in the motion compensation. For video encoder, the task of motion compensation is to find the best match from the reference frame as the prediction for the current coding unit, *i.e.*, macroblock (MB), by conducting motion estimation (ME) and then to decide the proper coding mode for this macroblock. The coding modes include intra-mode, inter-mode and different combination of block sizes used in the inter-mode. After motion estimation and mode decision, the difference between the current macroblock and its prediction as well as motion vectors (MV) and mode information will be compressed into the bitstream. In a video encoder, motion compensation consumes more than 50% computing power [33] and it is the most computationally intensive process. On the contrary, since all the information required for frame reconstruction is available in the bitstream, video decoder can directly reconstruct the current macroblock by adding the decompressed residue to its prediction block in the decoded reference frame indicated by the corresponding motion vectors. As a result, without conducting motion estimation and mode decision, the video decoder only requires 10%~20% computations of an encoder [33].

Due to the fact that video coding standards allow for optimizations in algorithms and implementations for motion compensation in the encoding stage, to design fast motion estimation and mode decision schemes in motion compensation is highly desirable for achieving a low complexity video encoder. Consequently, with reduced source coding complexity, the implementation of real-time video communication systems on those devices with limited computing power will become more feasible. Besides the motion compensation process, another important module in video encoder which has not been specified by the standard is the encoder bit-rate control. Rate control techniques have been widely used in compressed video communications. The purpose of rate control is to

regulate the output bit rates by adjusting the quantization parameters (QP) to achieve the best utilization of the available channel bandwidth. For real-time low-delay video communications, an efficient rate control scheme is indispensable to provide users with the best visual quality under the given channel condition. This research work mainly focuses on designing efficient algorithms for motion estimation, mode decision and rate control for video encoder.

1.2 Objective and Main Contributions

The primary objective of this work is to achieve better tradeoff between encoding complexity and quality as well as better utilization of the channel bandwidth. First, we investigate the most popular motion estimation technique: block matching algorithms (BMA). In motion compensated video coding, motion estimation plays an important part in reducing temporal redundancies between successive frames. The performance of a block matching algorithm in terms of speed and accuracy directly affects the complexity and the rate-distortion (R-D) performance of the video encoder. Our proposed BMAs aim at speeding-up the block matching process and achieving better R-D tradeoff. Second, we study one of the important mode decisions in inter-frame coding: variable size block motion estimation. The latest H.264/AVC standard employs seven different size block types in motion estimation. Our proposed fast inter mode decision scheme can save the computation cost significantly while keeping the R-D performance. Third, we investigate the difficulties involved in bit-rate control for H.264/AVC video coding. The aim of our proposed rate control scheme is to improve the accuracy in choosing frame level QPs and to adaptively allocate frame target bits according to its complexity. As a result, the visual quality fluctuation of reconstructed video frames is reduced. In our scheme, the most

essential part is the frame complexity estimation for both intra- and inter-coded frames.

The main contributions of this thesis are as follows:

- A thorough survey has been made for the existing block matching based motion estimation algorithms, mode decision schemes and various bit-rate control methods.
- We have developed two fast block matching algorithms for motion estimation. The proposed methods are based on the effective search pattern and partial distortion block matching criterion respectively. Both algorithms achieve improved tradeoff between estimation accuracy and complexity.
- We have developed a new measure of similarity between blocks in motion estimation called the smooth constrained mean absolute difference (SCMAD). Reduced bit rate for encoding the residue block can be achieved without any degradation of the reconstructed image quality.
- We have developed an efficient inter mode decision algorithm for H.264/AVC variable size block motion estimation. The proposed technique can save the computation cost significantly while maintaining the same R-D performance.
- We have proposed an adaptive intra-frame rate-quantization (R-Q) model for H.264/AVC video coding. The proposed model aims at selecting accurate QP for intra-coded frames. This model is based on our frame complexity estimation using frame-gradient as a measure.
- We have proposed a frame level target bits allocation scheme which takes into consideration both the buffer status and the predicted frame mean absolute difference (MAD) value. Based on the histogram information of successive frames, the MAD of current frame can be accurately estimated. Combined with the adaptive intra-frame R-Q model, when implemented on H.264/AVC video encoder, more accurate frame

level rate control is achieved. Besides, the overall quality for reconstructed images is improved with reduced quality fluctuations.

1.3 Outline of the Thesis

The thesis is organized as follows:

In Chapter 2, a detailed review of various existing techniques for block matching based motion estimation, mode decision and rate control is given.

In Chapter 3, we propose two fast block matching algorithms for motion estimation. Both methods are based on the observations of statistical characteristics of motion vector distribution. The first approach utilizes effective search pattern and search strategy to enhance the block matching accuracy with reasonable low computational complexity. The second approach achieves better complexity-quality tradeoff than conventional partial distortion based algorithms by improving prediction accuracy in a predictive search area while further reducing computations outside this area.

In Chapter 4, we present a novel and efficient criterion for block matching motion estimation. The proposed criterion uses a smoothness constraint on the residue block in block matching process. This criterion is more correlated with the number of actual bits generated for encoding the residue block. Therefore, better R-D performance can be achieved by coding the image at the same quality with reduced bit rate.

In Chapter 5, we investigate the variable size block motion estimation in H.264/AVC standard. The new standard employs 7 different size block types which can significantly improve the coding performance compared with previous video coding standards. On the other hand, the computational complexity increases dramatically due to the various

coding modes used. We propose a fast inter mode decision approach to efficiently reduce the number of candidate block types in motion estimation. The proposed algorithm can save the computation cost significantly while maintaining the same R-D performance.

In Chapter 6, we propose two frame complexity based schemes for H.264/AVC video rate control. The first scheme is an intra-frame R-Q model using gradient-based frame complexity measure. More accurate QPs for intra-coded frames can be determined using our proposed model. By considering both the buffer status and the predicted frame MAD in frame level target bits allocation, the second scheme aims at reducing the visual quality fluctuations. When both algorithms are employed for the H.264/AVC encoder, the overall performance of the rate control can be improved in terms of target bits mismatch and the visual quality of reconstructed video.

Finally, the conclusion and possible extension of this research are briefly addressed in Chapter 7.

Chapter 2

Overview of Motion Compensation and Rate Control Techniques

2.1 Typical Video Compression System

Digital video compression technologies have been growing rapidly in recent years due to the increasing demand of visual communication applications. Compared with audio signals, video signals require much more bandwidth for storage or transmission. Despite the development of storage technique and broadband networks, video compression is highly desirable for storing and transmitting video data. The video compression techniques are based on the reduction of both spatial redundancies and temporal redundancies within the source video data. The spatial redundancies are removed by intra-frame transform coding techniques while the temporal redundancies are eliminated by motion compensated inter-frame coding techniques. The hybrid coding methods containing both

Figure 2.1. Block diagram of a generic hybrid motion compensated video encoder

intra-frame and inter-frame coding schemes are most commonly used in the existing video coding standards such as H.261, H.262/MPEG-2, H.263, MPEG-1, MPEG-4 and the latest H.264/AVC.

Fig. 2.1 depicts a typical block diagram of a hybrid motion compensated video encoder. This coder can work on both intra- and inter-coding modes. For coding each frame, firstly the input original frame is segmented into blocks of 16×16 pixels, called the macroblocks (MB). Generally speaking, coding an intra MB does not depend on any other blocks (except for H.264/AVC, in which the intra prediction technique using neighboring blocks is enabled and only the intra prediction error is coded). The pixel values (or prediction error) of this intra MB will be directly passed to the discrete cosine transform (DCT) coding unit. In inter-coding mode, the current MB in the current frame is predicted by the most similar block in the reference frame with a displacement vector, *i.e.*, motion vector (MV). The process of finding the best matching block is called motion

estimation (ME). After motion estimation, the residue MB is formed by subtracting the current MB from the motion compensated block from the reference frame and this residue will go to the DCT coding stage. DCT coefficients are unequally quantized by a quantization matrix which is specially designed according to the spectral response of the human visual system (HVS). The quantized DCT coefficients together with the motion vectors are then encoded by an entropy coder. Note that, during the whole encoding process, the information loss only occurs in the quantization stage. The rate controller can determine how coarsely the DCT coefficients are quantized by indicating a quantization parameter (QP). The smaller QP will result in smaller distortion for the reconstructed frame and higher output bit rate for the encoder. On the contrary, using larger QP can achieve reduced bit rate with decreased video quality.

Figure 2.2. An example of frame configuration in a group of pictures (GOP)

Basically, there are three types of video frames depending on which prediction methods are used. In I (intra) frames, all the MBs are intra-coded without any motion compensation. In P (Predicted) frames, each MB can either be intra-coded or inter-coded

using motion estimation and compensation. The intra-/inter- decision is made based on the rule that the mode which can generate the smallest coding cost will be chosen. In B (bi-directionally interpolated) frames, each MB can be coded as an intra-block, an inter-block with forward prediction only, an inter-block with backward prediction only, or an inter-block which is bi-directionally predicted. Fig. 2.2 shows an example of the frame configuration in a group of pictures (GOP).

As mentioned in Chapter 1, although video coding standards have specified the implementation of video decompression process, the exact algorithms for implementing the encoder units have not been completely defined. In the following, more detailed descriptions will be given on the realization of motion compensation and rate control in a video encoder.

2.2 Block Matching Motion Compensation Techniques

The concept of using motion compensation to improve video compression efficiency can be traced back to early 1970s [29]. Based on the work in [65] and [66], various motion compensated coding systems have been developed. All of them follow the same concept which was stated in [65] that: *In an encoding system for use with video signals, the velocity of the subject between two frames is estimated and used to predict the location of the subject in a succeeding frame. As only the velocity and the updating difference information need to be transmitted, this provides a reduction in the communication channel capacity required for video transmission.* Fig. 2.3 shows an example of a motion compensated frame and its compensation residue. We can see from this figure that, after

Figure 2.3. Motion compensated frame and the corresponding residue frame. (a) Original current frame to be coded. (b) Difference between the current frame and its previous reference frame. (c) Motion compensated frame for frame (a) with corresponding motion vectors. (d) Prediction residue between the original frame and the motion compensated frame.

motion compensation, the difference between the motion compensated frame and its reference frame is much smaller than the difference between two original frames. As a result, transmitting the prediction difference and the prediction parameter information, *i.e.*, MV, is much more efficient than transmitting the original video frame data.

Generally speaking, to search for the predictions for the pixels in the current frame from a reference frame is conducted by the motion estimator and this process is called motion estimation. Motion estimation is mainly based on the spatial-temporal variation, namely, optical flow. Practically, most of the motion estimation algorithms assume the following conditions [10, 64]:

- 1) The illumination is uniform hence the observed object intensities are unchanged under movement.
- 2) Occlusion problem is neglected.

The first assumption neglects the change of illumination which may not correspond to a motion. The second assumption neglects the uncovered background problem since for the area of an uncovered background in the reference frame there are actually no optical flows.

Many motion estimation algorithms have been proposed in the past decades. They can be mainly categorized into three types:

- 1) Optical flow gradient based methods [67, 68]
- 2) Pel-recursive methods [69, 70]
- 3) Block matching methods [11]

Among the above methods, block matching techniques are very suitable in DCT based coding schemes due to its low computational complexity and good coding efficiency. Therefore, various video coding applications often depend on block matching motion compensation techniques. In the following section, we will discuss the block matching methods in more details.

2.2.1 Block Matching Method

Block matching is a technique that searches for the best match between the current block and the candidate blocks in a predefined area of the reference frame. Because of its low computational complexity and good performance, it is very suitable for both real-time software and hardware video coding applications. In block matching method, each block

Figure 2.4. Block matching motion estimation

is viewed as an independent object and we assume that all the pixels within a block have the same motion. First, the current frame is divided into small MBs with size $M \times N$, (most video coding standards adopt the MB size of 16×16) thus the block matching motion estimation is operated on MB level. Suppose that the maximum displacement (the motion vector MV) between the current MB and the candidate is $\pm d$ both horizontally and vertically. In other words, a search window of size $(2d + 1, 2d + 1)$ is defined. By calculating the matching criterion (similarity measure or block distortion measure) between the candidate and the current MB, the best matching candidate is found. The motion vector (u, v) is obtained from the offsets between the position of the best match and the position of the current MB. This procedure is depicted in Fig. 2.4.

Basically, there are three factors that determine the final performance of a block matching motion estimation algorithm in terms of accuracy and complexity:

- 1) Search range or search window size
- 2) Search pattern
- 3) Matching criterion

For the first factor, using a large search window will improve the matching accuracy. However, the computational complexity will increase rapidly with the search window size. For the second factor, the search pattern determines which candidate block in the reference frame should be checked. The full search (FS) method is the most straightforward block matching algorithm (BMA). It finds the best match by comparing all the candidates inside the search window. Its high computational complexity makes it not suitable for real-time applications. The selection of the block matching criterion also has a direct impact on the complexity and coding efficiency. In the following, we will discuss about different search and match strategies and block matching criteria in more details.

2.2.2 Fast Search Strategy for Motion Estimation

In the past two decades, many fast BMAs have been proposed to reduce the computation of motion estimation while maintaining acceptable matching accuracy. Most of them reduce the computation by limiting the number of checking points within the search window. In other words, they use different search patterns. They assume that the error surface decrease monotonically as the motion vector moves to the global minimum point [11]. This assumption is not always true in the real world since multiple local minima may exist in the search window. Thus it may lead those fast BMAs being trapped into local minima and result in poor matching accuracy. Some of the famous fast BMAs include three-step search (3SS) [12], 2D-logarithm search (LOGS) [11], orthogonal search (OSA) [13], cross search (CSA) [14], new three-step search (N3SS) [15], four-step search (4SS) [16], block-based gradient descent search (BBGDS) [17], diamond search (DS) [18, 19], and hexagon-based search (HEXBS) [20]. We take the 3SS, 4SS, DS and HEXBS as examples and describe them in the following. Other search strategies explor-

ing image features and motion properties can be found in the literature [9, 21, 22, 24, 34, 42, 43, 46].

Three-step search (3SS)

The three-step search algorithm was proposed by Koga *et al* [12] in 1981. The search pattern of 3SS is depicted in Fig. 2.5. The initial step size is equal to half of the maximum search range d i.e., $\lceil d/2 \rceil$. Taking search range $d=7$ as an example, for each step nine search points are compared and the winning point with the minimum block distortion (MBD) is chosen as the starting point in the following step. The step size is reduced by half after each step and the search ends when the step size is reduced to one. For $d=7$, totally $(9+8+8)=25$ checking points are needed. For larger search window, 3SS can be easily extended to a n -step search using the same strategy with the number of checking points equals to $[1 + 8\lceil \log_2(d+1) \rceil]$.

Figure 2.5. Three-step search: a search path for finding MV (5, -7)

Four-step search (4SS)

The four-step search algorithm (4SS) was proposed by L. M. Po and W. C. Ma [16] in 1996. This approach is more center-biased compared with 3SS. It uses a smaller initial step size than 3SS. As a result, the 4SS needs four searching steps to reach the boundary of a search window with $d=7$. The 4SS utilizes a halfway stop technique in its second and third step search. If the MBD point is found to be the central one, the step size is reduced by half and the process will jump to the fourth step. Fig. 2.6 shows two paths of 4SS. As can be seen from the figure, for the worst case (the upper-right path), it requires $(9+5+5+8)=27$ checking points. Generally, the algorithm can be extended as follows: if the step size of the fourth step is larger than one, another four-step search is performed with the first step equal to the last step of the previous search. The number of checking points required for the worst case is equal to $(9+18 \times \left\lceil \log_2 \frac{d+1}{4} \right\rceil)$.

Figure 2.6. Four-step search : two search paths for finding MV (7, -6) and MV (-7, 3)

Figure 2.7. Diamond search: a search path for finding MV (5, -2)

Diamond search (DS)

The diamond search (DS) [18, 19] also exploits the center-biased characteristics of the MVs in the real world video sequences. The DS algorithm employs two search patterns called large diamond search pattern (LDSP) and small diamond search pattern (SDSP). As can be seen from Fig. 2.7, the LDSP is used as the initial search step which is centered at the origin of the search window. After checking all the nine points of LDSP, if the MBD point is at the center of this LDSP then the search pattern in the following step will be switched to SDSP. Otherwise, the MBD point in the previous search step will become the new search center of the LDSP for the next search step. Finally, in the last search step, only four search points on the SDSP will be examined to determine which one is the best matching point. Fig. 2.7 depicts one example search path for finding MV (5, -2). Note that the search step using LDSP is unrestricted and the checking points outside the search window will be truncated.

Figure 2.8. Hexagon-based search: a search path for finding MV (5, -4)

Hexagon-based search (HEXBS)

The hexagon-based search (HEXBS) [20] employs a more circle-approximated pattern compared with the previous search patterns. Different from the LDSP pattern used in the DS algorithm, seven checking points instead of nine are used in the initialization step. These seven checking points consist of one center point at the origin surrounded by six endpoints of the hexagon. It can be observed from Fig. 2.8 that in the hexagon pattern the upper and lower edge points in the previous LDSP are eliminated while other 6 points are approximately uniformly distributed around the center. Similarly, the MBD point found in the previous search step will be positioned as the new search center for the following search step unless the MBD point is located at the hexagon center. In each movement of the hexagon, three new candidate points are checked and the search step using hexagon is also unrestricted until it reaches the search window boundary. Fig. 2.8 shows an example search path for finding MV (5, -4) in the HEXBS.

2.2.3 Block Matching Criteria

As mentioned in section 2.2.1, in block matching motion estimation, the first step is to segment a frame into blocks of size $(M \times N)$. Then the basic operation of block matching is to pick up a candidate block from the search range and calculate the matching criterion between the candidate and the current block. This procedure is repeated until all the candidates have been checked and finally the best matching one is selected. Here, the block matching is conducted in spatial domain and we can also operate in DCT domain in which the comparison is based on the DCT coefficients rather than the pixel intensities [25].

Assume that $x_t(k, l)$ is the intensity value of the pixel located at the upper-left corner of the block in the current frame t , and $x_{t-1}(k+i, l+j)$ to be the intensity value of the pixel shifted by i pixels and j lines within the search area in the reference frame, $t-1$. For the best match, the motion vector (MV_i, MV_j) indicates the displacement in horizontal and vertical directions, respectively. It is necessary to choose a proper matching function in the process of searching for the motion vector. Several matching criteria for BMA are summarized below:

- 1) Normalized Cross-Correlation Function (NCF) [26]

$$NCF(i, j) = \frac{\sum_{k=1}^M \sum_{l=1}^N x_t(k, l) x_{t-1}(k+i, l+j)}{\left[\sum_{k=1}^M \sum_{l=1}^N x_t^2(k, l) \right]^{1/2} \left[\sum_{k=1}^M \sum_{l=1}^N x_{t-1}^2(k+i, l+j) \right]^{1/2}} \quad (2.1)$$

$$(MV_i, MV_j) = (i, j) \mid \max NCF(i, j)$$

- 2) Mean Squared Error (MSE) [12]

$$MSE(i, j) = \frac{1}{M \times N} \sum_{k=1}^M \sum_{l=1}^N [x_t(k, l) - x_{t-1}(k+i, l+j)]^2 \quad (2.2)$$

$$(MV_i, MV_j) = (i, j) \mid \min MSE(i, j)$$

3) Mean Absolute Difference (MAD) [12]

$$MAD(i, j) = \frac{1}{M \times N} \sum_{k=1}^M \sum_{l=1}^N |x_t(k, l) - x_{t-1}(k + i, l + j)| \quad (2.3)$$

$$(MVi, MVj) = (i, j) \mid \min MAD(i, j)$$

4) Pel Difference Classification (PDC) [27]

$$T(k, l, i, j) = 1, \text{ if } |x_t(k, l) - x_{t-1}(k + i, l + j)| \leq \text{Threshold} \quad (2.4)$$

$$G(i, j) = \sum_{k=1}^M \sum_{l=1}^N T(k, l, i, j)$$

$$(MVi, MVj) = (i, j) \mid \max G(i, j)$$

5) Minimized Maximum Error (MiniMax) [28]

$$\text{Find } G(i, j) = |x_t(k, l) - x_{t-1}(k + i, l + j)|_{\max} \quad (2.5)$$

$$(MVi, MVj) = (i, j) \mid \min G(i, j)$$

For the above matching criteria, the NCF and MSE require enormous multiplications thus their hardware implementation cost is very high for real-time processing. Although PDC requires less hardware than MAD, its performance is quite sensitive to the threshold chosen. Compared with the traditional MAD criterion, the MiniMax criterion reduces hardware requirements at the cost of slightly degraded picture quality. So far the MAD criterion seems to be the most popular choice in designing practical image coding systems because of its good performance and relatively simple hardware requirements. We have carried out a set of experiments to compare the performance of the above matching criteria. Fig. 2.9 shows the resulting Mean Squared Error per frame for coding the *Foreman* sequence using different block matching criteria. From the figure we can find that using MSE as block matching function can achieve the smallest distortion for the reconstructed picture. The result of MAD is very close to MSE while outperforming all

Figure 2.9. Performance comparison of different block matching criteria when coding *Foreman* sequence. The frame Mean Squared Error for different criteria is as follows: MSE 36.13, MAD 37.42, NCF 41.59, PDC (th=20) 40.36, PDC (th=30) 42.71, PDC (th=40) 49.49, MiniMax 41.57.

the other criteria. Similar observations have also been reported in [29, 63]. Note that, besides using frame based MSE to evaluate the quality of a reconstructed video sequence, there is another commonly adopted function called the peak signal-to-noise ratio (PSNR) which is defined as follows:

$$PSNR = 10\log_{10}\left(\frac{255^2}{\frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N (x_{i,j} - \hat{x}_{i,j})^2}\right) \quad (2.6)$$

where M and N are dimensions of the picture and $x_{i,j}$ and $\hat{x}_{i,j}$ are the original and reconstructed pixel values respectively. Although the PSNR criterion is not a perfect measurement for the subjective quality test, it is currently widely used as an objective measurement for video coding performance evaluations due to its simplicity.

2.2.4 Fast Matching Strategy for Motion Estimation

In Section 2.2.2 we have studied one type of fast algorithms for motion estimation. These algorithms improve the search efficiency by restricting the number of candidates to be examined in the search window. Besides this type of search strategy, another approach for fast motion estimation which can be combined with full search (FS) method has also been studied extensively in recent decades. These fast FS algorithms are based on fast matching strategy in which the process of calculating block matching criterion between the current block and candidate block can be terminated in an early stage. In other words, only partial metric is computed when evaluating a candidate block [35]. When MAD is used as the block matching criterion, during the calculation of the sum of absolute difference (SAD), the accumulated partial SAD (PSAD) is used to eliminate the impossible candidate MVs. Thus, if the PSAD is larger than the current minimum SAD, this candidate is rejected without completing the remaining SAD computation. Many implementations of FS algorithms include this partial distortion search (PDS) technique to speed up the matching criterion computation.

Generally, we can classify these PDS approaches into two categories: *Adaptive matching scan* based methods [36, 37, 40, 41] and *Adaptive pixel decimation* based methods [23, 38, 44, 45]. In the first group, the order of partial distortion accumulation is reorganized in a way that those pixels with larger contribution to the total distortion measure will be used in the accumulation process first. In the second group, pixels in the MB are subsampled and grouped into a number of subblocks. The distortion accumulation is performed by gradually including more subblocks and comparing the accumulated distortion with an adaptive threshold. Much higher speed up gain can be achieved than the first group methods at the cost of increased probability of false rejection of candidates. Of particular relevance of our work are the adaptive pixel decimation based algorithms,

and which will be our focus in Chapter 3. In addition to the above mentioned fast FS algorithms using MAD as block matching criterion, other matching criteria such as integral projections (IP) can also be combined with FS [71].

2.2.5 Variable Size Block Matching Motion Compensation

The block matching technique originally introduced in [11] is widely adopted by various video coding designs mainly due to its particularly good tradeoff among simplicity, computational complexity and performance. In this scheme, each video frame is divided into a fixed number of square blocks, *e.g.*, MBs of size 16×16 , thus this scheme is normally called fixed size block matching (FSBM) motion compensation. FSBM-based approaches assume that in a video frame, each block of pixels will undergo the same translational motion. However, this assumption will not always hold since in a real-world video sequence the boundaries of the moving objects seldom coincide with the block boundaries. As a result, undesirable artifacts will be introduced when a block covers an area where two or more types of motion are present. In this case it is not possible to represent the complex motion within this block using only one motion vector. This can result in a large motion compensation error for such a block.

Many research proposals have been made to improve the FSBM scheme by varying the size of blocks to more accurately match moving areas. This type of strategies are generally referred to as variable size block matching (VSBM) algorithms. Chan *et al.* [62] introduced the VSBM concept in motion compensation schemes. Small blocks are used to cover areas of complex motion while regions of uniform motion, *e.g.*, the background of the scene, are covered by large blocks. Their method can adaptively divide the image into blocks of variable size to meet the assumption on uniform motion for all blocks [62].

Figure 2.10. Typical covering quadtree for VSBM [57]

Specifically, the frame partition algorithm partitions a frame into variable sized blocks using a binary tree approach. It begins with relatively large blocks, and these large blocks are further split into smaller blocks if the best matching error for using large blocks is greater than a threshold value. Such top-down splitting process continues until the maximum number of blocks has been obtained or the prediction error is below a certain threshold. More effective solutions to find out areas of uniform motion may be the quadtree-based approaches [57, 58, 59]. Rhee *et al.* in [57] proposed a bottom-up VSBM algorithm which relies more on local motion information. They detect areas of common motion, grouping them into variable sized blocks with a coding strategy based on the use of a covering quadtree. An example of a covering quadtree structure is shown in Fig. 2.10. A covering quadtree is a quadtree where each node has four children or none with [57]: 1) each node in the tree corresponding to a square block in the frame; 2) four child blocks or nodes of a block corresponding to the four equal-sized nonoverlapping square subblocks; and 3) the root node corresponding to the entire frame. The algorithm starts from

Figure 2.11. Block structures for FSBM and VSBM of *Stefan*.

searching a set of candidate motion vectors of each fixed-sized small block. Then the neighboring small blocks are merged in a quadtree manner if they have at least one vector in common. Although the optimal block size selection can be achieved using this bottom-up quadtree-based approach, it is very computationally demanding and not very suitable for real-time applications.

Proposals for employing VSBM have also been adopted in video coding standards such as H.263, MPEG-4 and the latest H.264/AVC. The H.263 and MPEG-4 standards were the first to allow using variable sized blocks. For example, in H.263 standard, four motion vectors can be used to represent the motions of corresponding 8×8 subblocks of the entire 16×16 MB. In other words, if the option of choosing four motion vectors is enabled, one MB can either be coded using 16×16 mode or 8×8 mode. The one/four motion vectors decision is made depending on which mode can have minimum coding cost. In addition to the above mentioned two modes, five other block sizes are also enabled in the H.264/AVC standard including 16×8 , 8×16 , 8×4 , 4×8 and 4×4 . Fig. 2.11 depicts an example of block structures using FSBM and VSBM by an H.264/AVC

video encoder. In Fig. 2.11(a), the block size is fixed to 16×16 while in Fig. 2.11(b) all seven block types are allowed to be chosen. It can be seen that compared with the fixed size block partition, smaller sized blocks are adaptively employed in the VSBM scheme to cover the moving object, *i.e.*, the tennis player, in the scene. Note that, in order to find the best mode the initial procedure defined in H.264/AVC standard is to exhaustively examine all partition modes and finally choose one with the minimum Lagrangian cost [52, 53, 54]. Many fast mode selection strategies have been proposed [47, 48, 49, 50, 60, 61] to reduce the computational burden of H.264/AVC and further investigations will be provided in Chapter 5.

2.3 Rate Control for Video Coding

Rate control plays an important role in compressed video communications. It regulates the output rate of the coded bitstream in order to achieve the best video quality under the channel bandwidth constraint. Although video coding standards have defined the syntax of video decoding, the issue of designing the rate controller for an encoder is left open for optimizations. In this section, we will give a review for some basic concepts in rate control and show some examples of the existing standard rate control algorithms for video coding.

2.3.1 Quantization Process

In a typical DCT-based video encoder, after motion compensation or intra-prediction, the residue signal in spatial domain will be converted into transform domain coefficients using traditional 2D-DCT. These DCT coefficients are further quantized by a quantizer. In general, a quantizer maps a signal with a large range of values to a quantized signal

Figure 2.12. Relation between QP and Qstep

with a reduced range of values. The quantized signal contains less information than the signal before quantization thus it can be coded using fewer bits. Normally, the video encoder employs a scalar quantizer to map one sample of input signal to one quantized output by taking the following operation:

$$Z_{i,j} = \text{round}(Y_{i,j} / Q_{\text{step}}) \quad (2.7)$$

where $Y_{i,j}$ is a DCT coefficient, Q_{step} is a quantization step size (or quantization scale) and $Z_{i,j}$ is the quantized coefficient. The rounding operation can be generally regarded as rounding to the nearest integer.

After the DCT transformation and quantization, the high frequency coefficients will be most likely removed since they are usually quantized to zeros and the remaining nonzero coefficients will represent the low frequency information. As the quantization operation is not reversible, the lost information can not be recovered by the inverse quantization. The critical factor determining how coarsely the transformed coefficients will be quantized is the quantization step size Q_{step} . Higher compression ratio can be achieved if a larger Q_{step} is used during quantization and vice versa. Meanwhile, using a larger Q_{step} can result in increased information loss and increased coding distortion.

Basically, in video coding system another factor called quantization parameter (QP) is applied to determine the Q_{step} . In general, the larger the QP the larger the Q_{step} . The relation between QP and Q_{step} is shown as in Fig. 2.12. In previous video coding standards such as H.263, the relation between QP and Q_{step} is usually linear as illustrated in Fig. 2.12 (a). However, the latest H.264/AVC standard adopts a non-linear relation between QP and Q_{step} in its quantization scheme as shown in Fig. 2.12 (b). This change is mainly because of the advanced integer transform and division-free quantization scheme in H.264/AVC standard [54, 56]. To achieve the goal of regulating the output bit rate of an encoder, the rate controller should be able to adjust the QP (or Q_{step}) adaptively according to the channel and buffer conditions.

2.3.2 Variable Quantization Scale Rate Control

As described before, video compression is achieved by reducing both the temporal redundancies and spatial redundancies within the source video data. Since in real-world video sequences these two types of redundancies are time-varying, therefore if no rate control mechanism is employed at the source coder side, *i.e.*, a fixed QP is used during the encoding process, the output bit rate of the compressed bitstream will also vary in time. For example, the output bit rate will increase at the scenes which contain complex motions (with higher temporal activity) while fewer bits will be generated when coding more stationary scenes (with lower temporal activity). This relationship between temporal activity and output bit rate is illustrated in Fig. 2.13. As can be seen in Fig. 2.13 (b), if a constant QP is used throughout the coding process, large fluctuations in the output bit rate will be observed while constant visual quality can be achieved for the compressed video. As the output bit rate to the channel is varying, this kind of coding scheme is referred to as variable bit rate (VBR) encoding. The VBR video encoders are suitable for the

Figure 2.13. VBR and CBR coding schemes with corresponding relations between the output bit rate and quality.

applications where very large buffers and fast channels are available. However, for real-time low-delay video communications, the constant bit rate (CBR) encoding is more desirable as depicted in Fig. 2.13 (c). In CBR scheme, the buffers have to be put between the encoder and the CBR channel to smooth out the variation of the output compressed bitstream. Given the buffer and channel conditions, the rate controller should adaptively change the QP for the current coding unit (a frame or a MB) such that the bit rate of the compressed video data can comply with the channel bandwidth. Moreover, the buffer fullness should be carefully controlled to avoid overflow and underflow. Because the

undesirable frame skipping will occur in case of buffer overflow which will cause motion jerkiness and on the other hand the channel bandwidth is wasted when buffer underflow happens. Besides regulating the output bit rate according to the buffer and channel status, another objective of rate control is to achieve stable and better reconstructed video quality by adaptively allocating bit budget for different coding units.

Normally, the rate control process can be segmented into the following stages as shown in Fig. 2.14. Firstly, the target bit allocation for the current coding unit is performed. In this stage the coding unit can be a GOP in the video sequence, a frame in the current GOP or a MB in the current frame. The bit allocation can be done not only based on the channel bandwidth and the current buffer status but also by taking the temporal complexity into consideration [85, 86, 91, 93]. As a result, more consistent video quality can be achieved with less quality fluctuation. In addition, region based bit allocation

Figure 2.14. Procedures for a typical video rate control scheme

methods [102, 103, 104] can allocate more bits to the region-of-interest (ROI) within a video frame such as the foreground while coding the background with fewer bits in order to achieve improved subjective visual quality. Secondly, after bit allocation for the current coding unit, the rate controller should choose an appropriate QP for the quantizer using a certain rate-distortion (R-D) model. Ideally, if we know the exact R-D characteristics of each coding unit we can select the optimal QP thus the output bit rate of this coding unit will match its target bit budget perfectly. However, since the actual R-D

characteristics are not available without any pre-encoding process, mathematical R-D models are widely used to estimate the QP based on previous coding statistics [81, 83, 98, 105, 106, 107]. Thirdly, the current coding unit will be coded using the QP determined by the R-D model. Finally, the R-D model and buffer status are updated according to the actual coding statistics. In the following we will introduce some of the existing rate control schemes to further illustrate the implementation of the rate controller in a video encoder.

2.3.3 Standard Rate Control Algorithms

Various rate control techniques have been proposed for video coding in the past decade. In this section, we provide a brief review of three well-known rate control algorithms that have been adopted by the international standards including TMN8 [74, 108] for H.263, VM18 [75] for MPEG-4 and adaptive rate control algorithm [77, 78] for H.264/AVC.

TMN8 Rate Control Algorithm

Generally, the TMN8 [74] rate control is performed at two layers: frame layer target bit allocation and MB layer QP selection. In the first step, the number of bits allocated to the current frame is determined based on the current buffer fullness, the frame rate and the channel bandwidth. The current buffer fullness can be calculated as

$$W = \max(W_{prev} + D - R / F, 0) \quad (2.8)$$

where D is the actual number of bits used for encoding the previous frame, W_{prev} is the previous buffer fullness, R is the channel bandwidth and F denotes the frame rate. Similar to TMN7 [111] rate control scheme, the TMN8 method skips encoding next frames when the buffer fullness W is larger than a predefined threshold M , ($M = R / F$ by default for low-delay applications). Note that, this frame skipping method is to

guarantee that the buffer delay will not be larger than a frame interval. The frame bit budget for the current frame is further estimated as

$$B_r = \frac{R}{F} - \Delta \quad (2.9)$$

where

$$\Delta = \begin{cases} W / F, & W > Z \cdot M \\ W - Z \cdot M, & \text{otherwise} \end{cases} \quad (2.10)$$

in which by default Z is set to 0.1. The MB layer rate control aims at selecting QP for each MB within the current frame after the frame layer bit allocation is done. Based on the R-D model proposed in [89, 94], the optimal quantization step size Q_i^* for the i th MB in the current frame is derived by the following equation

$$Q_i^* = \sqrt{\frac{256K_i}{L_i} \cdot \frac{\sigma_i}{\alpha_i} \cdot S_i} \quad (2.11)$$

where σ_i is the standard deviation of the luminance and chrominance values in the MB, $S_i = \sum_{k=i}^N \alpha_k \sigma_k$ is the weighted energy sum of the remaining MBs in the current frame. $L_i = \beta_i - 256N_i C_i$ is the number of remaining bits for coding the rest MBs in the current frame, excluding the header bits. Note that, β_i and N_i represent the bits budget and the number of MBs that remained in the current frame respectively. The model parameters K_i and C_i are updated adaptively after coding each MB [74]. The TMN8 rate control algorithm performs better than the previous approaches for H.263 video in terms of the reconstructed video quality and the bit rate control accuracy.

VM18 Rate Control Algorithm

The VM18 [75] rate control algorithm designed for MPEG-4 video coding is based on the quadratic R-D model [81, 82] given by

$$R_n = S_n \left(\frac{X_{1,n}}{Q_n} + \frac{X_{2,n}}{Q_n^2} \right) \quad (2.12)$$

where Q_n is the quantization step size of the current frame n , S_n denotes the encoding complexity which is often substituted by the MAD of the residue frame. Based on the assumption that neighboring video frames of the same type may have similar R-D characteristics, the model parameters $X_{1,n}$ and $X_{2,n}$ are updated by linear regression method from previous parameters.

The target bit rate for the current frame is computed based on the bits available and the bits of the last encoded frame. Basically, if the bit rate of the previous frame is high, more bits will be allocated to the current frame. On the other hand, if there are fewer bits left for encoding, fewer bits will be assigned to the current frame. The initial target bit rate is calculated by

$$T_n = \text{Max} \left(\frac{R_s}{30}, 0.95 \frac{R_r}{N_r} + 0.05 R_{n-1} \right) \quad (2.13)$$

where R_s is the bit rate for the sequence, R_r is the number of remaining bits for encoding this sequence, N_r is the number of remaining frames to be encoded and R_{n-1} is the actual bit rate for coding the previous frame. The initial frame bit budget from (2.13) will be further adjusted to avoid the buffer overflow and underflow [75]. When frame layer bit allocation is completed, the QP for the current frame can be computed using (2.12) and it will be clipped between 1 and 31. After encoding the current frame, if the buffer status is above 80%, the next frame is skipped in order to prevent the buffer from overflowing.

Adaptive Rate Control for H.264/AVC

The concept of *basic unit* was introduced for H.264/AVC video rate control in [77, 78]. In this framework, the rate control can be implemented at different levels according to the definition of the basic unit such as frame level, slice level, MB level or a group of contiguous MBs. Similar to previous rate control algorithms, the target bit allocation is conducted based on both the channel bandwidth and the buffer status. The major differ-

ence between the H.264/AVC rate control scheme and previous methods is that the QP for the current coding unit has to be determined before the motion compensation process in H.264/AVC video coding. Since the actual statistics of the residue are not available before the QP calculation, this results in a typical *chicken and egg* dilemma [77]. In order to resolve this problem, the adaptive rate control algorithm utilizes a linear prediction for the characteristics of current frame based on the previous frame. Although this method works well for relatively stationary scenes or frames which contain slow motions, its performance will suffer from complex motions and scene changes. More details on the H.264/AVC rate control and our proposed frame complexity based scheme will be introduced in Chapter 6.

2.4 Summary

In this chapter, we have reviewed the basics of motion compensation and rate control techniques for a generic video compression system. The motion compensation method is very essential in reducing the temporal redundancies among source video frames while rate control schemes are employed to regulate the output bit rate of the encoder to comply with the channel bandwidth. Both technologies are indispensable for the efficient video communication applications. In the subsequent chapters, we will discuss our proposed approaches in motion compensation and rate control and comparisons between different schemes will be given.

Chapter 3

Fast Block Matching Motion

Estimation Algorithms

3.1 Introduction

As discussed in Chapter 2, in order to reduce the computational complexity of full search (FS), many fast block matching algorithms have been developed in the past decades. These algorithms can be classified into two categories: 1) fast search algorithms which restrict the number of candidates to be examined in the search window and 2) fast matching algorithms which are based on partial distortion search (PDS) strategy.

In this chapter, we develop two simple and effective fast block matching approaches for motion estimation. The first approach is an efficient three-step search (E3SS) algorithm which belongs to the fast search category. The traditional three-step search (3SS) [12] algorithm has been widely used in block matching motion estimation due to its simplicity

and effectiveness. Searching with a large pattern in the first step, 3SS is more efficient to find the global minimum especially for those sequences with larger motions. However, for stationary or quasi-stationary blocks it will easily result in the search being trapped into a local minimum. The proposed E3SS modifies the 3SS by employing a center-biased small diamond pattern in the first step and the unrestricted search strategy is used to search for the center area. Thus it can find the small motion vectors more effectively.

Our second method belongs to the fast matching category. It is an improved partial distortion search algorithm with a predictive search area. This novel partial distortion method can significantly enhance the overall performance of the traditional adjustable partial distortion search (APDS) [38]. The proposed algorithm is based on the observation that when the normalization is employed for partial distortion calculation, the false elimination of candidates will be more likely to happen near the true motion vector in the search window. In order to further speed up the conventional APDS while maintaining the same quality, in the proposed method we adaptively adopt the quality factor in a predictive search area in the search window in which the erroneous rejection of candidates occurs most frequently. The determination of such predictive search area is based on the spatial correlation of neighboring motion vectors. Simulation results show that our proposed method performs much better than the other PDS based fast full-search methods. Moreover, substantial speed-up gain can be achieved with negligible prediction degradation compared to FS.

The rest of this chapter is organized as follows. In Section 3.2, we will present in details the development of our proposed efficient three-step search algorithm. In Section 3.3, our novel partial distortion search method using predictive search area is discussed. The remarks on both types of fast block matching motion estimation algorithms are given in Section 3.4. Finally, we draw a conclusion in Section 3.5.

Figure 3.1. Motion vector distributions of (a) *Football* and (b) *Foreman*

3.2 Efficient Three-Step Search Algorithm

3.2.1 Observations

In real-world video sequences, the distribution of motion vectors is highly center-biased. Fig. 3.1 shows the motion vector distributions of two sequences. More about the characteristics of motion vector distribution can be found in [29]. As can be observed in Fig. 3.1, for *Football* and *Foreman* sequences which contain large motions, the distributions of motion vectors are still highly center-biased. There are about 60% and 85% motion vectors located within the central 5×5 area for *Football* and *Foreman* respectively. In order to take advantage of this characteristic, the new three-step search (N3SS) [15] modified the checking points pattern of 3SS in the first step by searching additional eight points in the center. In the mean time, N3SS employed a halfway-stop technique to speed up the (quasi-)stationary blocks matching. Experimental results show that the N3SS performs better than 3SS in terms of mean square error and the number of search points. Another BMA using center-biased search pattern is four-step search (4SS) [16] which utilizes a smaller 5×5 grid in the first step. As a result, 4SS requires four steps to reach

Figure 3.2. Multiple local minima within a search window

the boundary checking points for the search window size $w = 7$. The performance of 4SS is better than the 3SS and comparable to the N3SS with fewer search points. In fact, all of the fast BMAs are based on the assumption [11] that the block distortion measure (BDM) increases monotonically as the checking point moves away from the global minimum block distortion (MBD) point. However, this is not always true in the real world especially for large motion blocks where multiple local minima may exist in the search window. In Fig. 3.2 we depict an error surface of a block from *Tennis* sequence which contains multiple local minima within a search window. When a large search window is used *e.g.*, $w = 15$, the initial step size for the above three algorithms will increase by a factor of 2. Therefore, the large 17×17 grid and 9×9 grid with the nine sparsely distributed checking points in the first step of 3SS and 4SS are more likely to mislead the search to a wrong direction. In addition, the halfway-stop technique in N3SS will lose its accuracy as well for it may miss the chance to find the global minimum surrounding the central 5×5 area. In the following, we will briefly review the traditional N3SS before the development of our proposed E3SS algorithm.

Figure 3.3. Two different search paths for finding MV within 5×5 area in N3SS

3.2.2 Review of New Three-Step Search Algorithm

For the search window $w=7$, in the first step the N3SS algorithm utilizes a 3×3 grid search pattern at the center in addition to the larger 9×9 grid in the 3SS. If the MBD point is found at the center of the search window, the search will stop. If the MBD point is one of the eight points on the 3×3 grid, only additional five or three points will be checked, otherwise the search window center will be moved to the winning point and the following procedure is the same as in 3SS. The details are as follows:

- Step 1: Totally $8+9=17$ points are checked including the central nine points on the 3×3 grid and the eight neighboring points on the 9×9 grid. If the MBD point is the search window center, the search will be terminated; otherwise go to step 2.
- Step 2: If one of the eight neighboring points on the 3×3 grid is found to be the minimum in the first step, go to step 3; otherwise go to step 4.

- Step 3: Move the small 3×3 search window so that the window center is the winning point found in Step 1. Search additional five or three points according to the location of the previous MBD point, then the search will stop.
- Step 4: Reduce the large 9×9 search window size by half and move the center to the MBD point in Step 1, follow the searching process of Step 2 and Step 3 in 3SS.

Fig. 3.3 shows two different search paths for finding motion vectors within a 5×5 area. According to the halfway-stop technique, the N3SS needs 17 search points for stationary blocks and 20 or 22 points for small motion vectors within the central 5×5 area. In the worst case, $25 + 8 = 33$ search points will be required compared to 25 points needed in 3SS.

3.2.3 Development of Efficient Three-Step Search Algorithm

In order to exploit the center-biased characteristics of motion vector distribution in real-world video sequences, a small diamond search pattern (SDSP) [18, 19] is utilized in the search window center. Fig. 3.4 shows the search pattern used in the first step of our E3SS, we take the search window $w = 7$ as an example. Thus, in the first step, totally $9 + 4 = 13$ points will be checked which is four points more than 3SS and four points fewer than N3SS. If the winning point is the search window center the search will be stopped. If the MBD point is one of the eight neighboring points on the large 9×9 grid, the following process will be the same as in 3SS. If the minimum is one of the four points on the small diamond, the center of the small diamond is set to the minimum point and another three points will be checked. The center of the small diamond is moved to the winning point each time until the minimum is found in the center. The E3SS differs from N3SS in that 1)

Figure 3.4. Search pattern used in the first step of E3SS

a small diamond pattern is used instead of a square pattern in the central area and 2) unrestricted search step for the small diamond rather than a single movement for the small square. The E3SS can be summarized as follows: (taking search window $w=7$ as an example)

- Step 1: In addition to the nine checking points in 3SS, four points on a small diamond in the center are added. If the MBD point is in the search window center, the search will be stopped. Otherwise, go to Step 2.
- Step 2: If the MBD point in Step 1 is one of the eight neighboring points on the 9×9 grid, the rest search procedure is the same as in 3SS. Otherwise, go to Step 3.
- Step 3: Move the small diamond so that its center is the previous winning point. Continue to search the other points on the small diamond. There is no restriction on the search step unless the MBD point is the small diamond center or

the small diamond reaches the boundary of the search window. More importantly, only three or two extra points are checked each time.

Fig. 3.5 shows the block diagram of the E3SS algorithm. In Fig. 3.6 we illustrate a search path for finding motion vector $(3, -1)$ when the winning point in the first step is $(1, 0)$. It is obvious that when the MBD point is found in one of the neighboring points around center $(0, 0)$ in the first step, utilizing unrestricted search step can increase the probability to find the true motion vector which located near the central 5×5 area. In general, the E3SS needs 13 points for stationary blocks and 16 to 21 points for small motion vectors within the central 5×5 area compared with 17, 20 or 22 points needed in N3SS respectively. For the worst case, $9 + 4 + 8 + 8 = 29$ points are required while 25, 33 and 27 points are needed in 3SS, N3SS and 4SS respectively.

Figure 3.5. Block diagram of the E3SS algorithm

Figure 3.6. Search path for finding MV (3, -1) in E3SS

3.2.4 Simulation Results

The experiments are conducted by using the luminance component of five video sequences *Coastguard* (CIF 352×288, 300 frames), *Football* (CCIR601 720×486, 59 frames), *Suzie* (CIF 352×288, 100 frames), *Mother & Daughter* (CIF 352×288, 100 frames) and *Salesman* (CIF 352×288, 100 frames). The first two sequences contain medium to large motion activities while the others consist of relatively small motions. The distortion measurement of mean absolute difference (MAD) is used as the BDM. The search window $w = 7$ and $w = 15$ are both used for a MB size of 16×16 .

Three criteria are used for comparing our E3SS against five other BMAs—FS, 3SS, 4SS, N3SS and DS. They are mean square error (MSE) between the estimated frames and the original frame, the average number of search points per MB (SearchPT), and the speed up ratio with respect to FS (Speed-up). The experimental results for comparison are summarized in Table 3.1 to Table 3.5. It can be seen from Table 3.1 and Table 3.2 that for sequences which contain medium to large motions and regardless of search window size the proposed E3SS gives the smallest MSE among the five fast algorithms while its

Table 3.1 Performance comparisons for CIF sequence *Coastguard*

BMA	Window size w=7			Window size w=15		
	MSE	SearchPT	Speed-up	MSE	SearchPT	Speed-up
FS	63.20	225.00	1.00	61.87	961.00	1.00
3SS	67.26	25.00	9.00	73.67	33.00	29.12
4SS	66.97	19.45	11.57	67.41	27.62	34.79
N3SS	64.90	21.27	10.58	71.86	21.09	45.57
DS	66.11	13.55	16.61	65.45	13.64	70.45
E3SS	64.22	13.82	16.28	64.05	13.88	69.23

Table 3.2 Performance comparisons for CCIR601 sequence *Football*

BMA	Window size w=7			Window size w=15		
	MSE	SearchPT	Speed-up	MSE	SearchPT	Speed-up
FS	257.83	225.00	1.00	145.06	961.00	1.00
3SS	276.22	25.00	9.00	186.04	33.00	29.12
4SS	289.72	20.56	10.94	193.54	29.80	32.25
N3SS	276.70	23.81	9.45	192.82	25.68	37.42
DS	289.30	20.72	10.86	217.97	22.75	42.24
E3SS	275.59	20.29	11.09	182.47	22.63	42.47

computational complexity is less than 3SS, 4SS, N3SS and comparable to DS. Table 3.3 to Table 3.5 show that the E3SS is also suitable for searching small motion vectors. It achieves similar or smaller MSE than that of N3SS and DS while it needs one more search point compared with DS. On average the E3SS requires three to four points fewer than the N3SS thus a 15% speed up ratio against N3SS can be achieved. In addition, we can find that when using large search window, the search points needed for 3SS and 4SS increased dramatically due to the increasing search step while the computational requirement for E3SS, DS and N3SS only changed slightly. Thus, the E3SS is more robust than the other fast BMAs like 3SS and 4SS. Fig. 3.7(a) and Fig. 3.7(b) illustrate the frame by frame comparison of E3SS, 3SS, 4SS, N3SS and DS in terms of MSE and the average

number of search points per block respectively for the *Football* sequence using search window $w = 15$.

Table 3.3 Performance comparisons for CIF sequence *Suzie*

BMA	Window size $w=7$			Window size $w=15$		
	MSE	SearchPT	Speed-up	MSE	SearchPT	Speed-up
FS	11.89	225.00	1.00	11.82	961.00	1.00
3SS	12.52	25.00	9.00	12.71	33.00	29.12
4SS	12.29	17.37	12.95	12.37	25.51	37.67
N3SS	11.93	17.96	12.53	12.00	17.92	53.63
DS	12.05	13.65	16.48	12.03	13.65	70.40
E3SS	12.07	13.95	16.13	12.08	13.91	69.09

Table 3.4 Performance comparisons for CIF sequence *Mather&Daughter*

BMA	Window size $w=7$			Window size $w=15$		
	MSE	SearchPT	Speed-up	MSE	SearchPT	Speed-up
FS	17.63	225.00	1.00	16.02	961.00	1.00
3SS	18.45	25.00	9.00	17.70	33.00	29.12
4SS	18.31	17.97	12.52	17.24	26.29	36.55
N3SS	17.88	19.36	11.62	16.97	19.43	49.46
DS	18.04	14.45	15.57	16.99	14.51	66.23
E3SS	18.08	15.60	14.42	16.94	15.79	60.86

Table 3.5 Performance comparisons for CIF sequence *Salesman*

BMA	Window size $w=7$			Window size $w=15$		
	MSE	SearchPT	Speed-up	MSE	SearchPT	Speed-up
FS	18.82	225.00	1.00	18.79	961.00	1.00
3SS	19.70	25.00	9.00	19.96	33.00	29.12
4SS	19.45	17.40	12.93	19.69	25.49	37.70
N3SS	18.94	18.19	12.37	19.25	18.15	52.95
DS	19.41	13.85	16.25	19.41	13.85	69.39
E3SS	18.95	14.23	15.81	19.00	14.23	67.53

(a)

(b)

Figure 3.7. Frame by frame comparisons for *Football* sequence (720×486 , 59 frames) with search window $w = 15$. (a) MSE per frame. (b) Average search points per MB

3.3 Partial Distortion Search Using Predictive Search Area

Instead of reducing the number of searching points in the search window, the concept of PDS is that when calculating the block matching measure of a candidate, if the partial measure is found greater than the total measure of the current minimum vector then the calculation will be stopped, and this candidate will be rejected without completing the rest of the computation in the block matching measure. Different from the lossless PDS methods which we classified as adaptive matching scan based methods in Section 2.2.4, the normalized partial distortion search (NPDS) [23] normalizes both the accumulated partial distortion and current minimum distortion before comparison. More computations are saved at the cost of slightly reduced visual quality. More recently, the adjustable partial distortion search (APDS) [38] is proposed to further improve the speed-up performance of NPDS by introducing a few more progressive partial distortions (PPDS) in the first stage. In addition, a quality factor k is employed to reduce the probability of erroneous eliminations. Based on the investigation of the characteristics of the APDS in the following, our proposed improved method will be developed step by step.

3.3.1 Traditional Adjustable Partial Distortion Search

Instead of calculating the total distortion consisting of all pixels of a 16×16 MB at one time, in NPDS and APDS, a grouping method is applied to divide pixels in one MB into 16 groups with evenly distributed patterns as shown in Fig. 3.8. As a result, the sum of absolute differences (SAD) between current MB and a candidate MB is subdivided into 16 partial distortions d_p where the p th partial distortion is given as

Figure 3.8. Pixel grouping pattern for calculation of the partial distortions

$$d_p = \sum_{i=0}^3 \sum_{j=0}^3 |I_t(4i + s_p, 4j + t_p) - I_{t-1}(4i + s_p + u, 4j + t_p + v)| \quad (3.1)$$

where $1 \leq p \leq 16$ and the p th accumulated partial distortion SAD_p is defined as

$$SAD_p = \sum_{i=1}^p d_i \quad (3.2)$$

Note that, in the above equations, $I_t(\cdot, \cdot)$ and $I_{t-1}(\cdot, \cdot)$ denote the pixel values in the current frame and previous reference frame respectively. (u, v) denotes the candidate motion vector and (s_p, t_p) indicate the offsets of the upper left corner point of the p th partial distortion from the upper left corner of the MB. The order of calculation of the 16 partial distortions is illustrated as in the upper left part of Fig. 3.8. In order to further increase the speed-up ratio of NPDS, the first partial distortion of NPDS is replaced by several smaller progressive partial distortions (PPD) in APDS. As an example, the bottom left corner of Fig. 3.8 shows that the pixels involved in the first partial distortion are further divided into four groups with (q_h, r_h) denoting the offsets of upper left corner of the h th PPD from the upper left corner of d_1 . Similarly, the h th partial distortion in d_1 is defined as

$$d_h = \sum_{i=0}^1 \sum_{j=0}^1 |I_t(8i + 4q_h, 8j + 4r_h) - I_{t-1}(8i + 4q_h + u, 8j + 4r_h + v)| \quad (3.3)$$

where $1 \leq h \leq 4$. As a result, the total distortion of one candidate MB is calculated by $16 + 4 - 1 = 19$ stages. To reject a candidate when we have calculated the g th ($1 \leq g \leq 19$) accumulated partial distortion SAD_g , the normalized comparison is performed to test whether $N^2 SAD_g > f(n, k) SAD_{MIN}$ is satisfied while $N = 16$ represents the MB size and SAD_{MIN} is the current minimum distortion. The adjustable function $f(\cdot)$ with a quality factor $k \in [0, 1]$ is designed to replace the original n which indicates the number of pixels used in the current accumulation. It is defined as

$$f(n, k) = (1 - k)n + kN^2 \quad (3.4)$$

Therefore, with the quality factor k changing from 0 to 1, APDS could perform as the original NPDS with PPD and the conventional PDS.

3.3.2 Predictive Search Area Based Partial Distortion Search

The efficiency of a partial distortion based fast full-search algorithm depends heavily on the speed to reject invalid candidates and how fast the block distortion measure can be calculated. As can be seen from previous descriptions, the speed-up improvement of APDS over NPDS comes from the fact that four smaller partial distortions are employed to replace the original first partial distortion. Thus theoretically, when rejecting one impossible candidate, the maximum computational reduction ratio of NPDS over FS has been increased from 16 to 64. On the other hand, the fewer the number of pixels used in the normalized comparison the higher the probability of erroneous rejections. In order to deal with this potential problem, the adjustable function $f(n, k) = (1 - k)n + kN^2$ can arbitrarily increase the number of pixels used in the current normalized comparison. In other words, the approximated total distortion of current candidate has been reduced to

Figure 3.9. Distribution of differential MVs

achieve more accurate evaluation at later stages. And it is clear that later rejection implies higher resultant quality [38].

Although this arbitrary adjustment can significantly reduce the probability of false elimination, its advantage has not been utilized optimally. In order to illustrate this phenomenon, various kinds of standard test video sequences are used to generate the true motion vectors found by original FS and the motion vectors found by APDS without using the adjustable function *i.e.*, ($k = 0$). Fig. 3.9 demonstrates the probability distribution plot of the differential motion vectors. As can be seen from Fig. 3.9, the distribution of differential motion vectors is highly center-biased. That is to say, when normalized partial distortion is employed, false elimination of candidates occurs most likely within a small area near the true motion vectors. Furthermore, from our observations, the probabilities for differential MVs locating in a $\pm D$ square window are as follows: 75.67% for $D = 0$, 94.18% for $D = 1$, 96.46% for $D = 2$, and 97.94% for $D = 3$ and so on. Such

Figure 3.10. Locations of neighboring MBs

highly center-biased characteristic implies that if the arbitrarily adjustable function f is only employed for candidates in a predictive search area adjacent to the true motion vector, the computations of APDS can be greatly reduced while maintaining the same prediction accuracy.

As mentioned above, the predictive search area is a small search window around the true motion vector. However, the actual location of true motion vector is not available until full search is conducted in motion estimation. Since in real-world video sequences, the motion of an object has high correlation in the spatial domain. As a result, the information of previously coded neighboring motion vectors can be used to predict the motion vector for the current MB. In our proposed method, the median of three neighboring motion vectors corresponding to the MBs, one to the left (A), one above (B) and one above right (C) is used as the predictor:

$$MV_{pred} = median(MV_A, MV_B, MV_C) \quad (3.5)$$

The locations of neighboring MBs are illustrated in Fig. 3.10. Note that there are cases when some or all neighboring MBs are outside the picture boundary. In such case, the unavailable neighboring motion vectors will be replaced by (0,0). The reason to choose median motion vector as our predictor is its simplicity yet high accuracy [30, 39].

In addition, when the predicted motion vector is available for the current MB, it will be used as the new starting point for the motion estimation. As illustrated in Fig. 3.11, the

Figure 3.11. Formation of the predictive search area (PSA) around the predicted MV

original search window center is shifted to a new starting point and our predictive search area (PSA) is the grey square area with size $\pm D$ around the predicted motion vector. The proposed algorithm can be summarized as follows:

- Step 1: Determine the predictive search area (PSA) in the new search window according to the predicted motion vector MV_{pred} . The search begins at the new starting point and then move outward in a spiral-scanning path.
- Step 2: For a new candidate MB, initialize $g = 1$, $SAD_0 = 0$. Enable k if the candidate is in the predictive search area, otherwise $k = 0$.
- Step 3: Calculate the g th ($1 \leq g \leq 19$) accumulated partial distortion SAD_g for the candidate according to (3.1) (3.2) (3.3).
- Step 4: Verify whether $N^2 SAD_g > f(n, k) SAD_{MIN}$ is satisfied. If it is satisfied, reject the current candidate, move to a new candidate and go to Step 2. Otherwise, go to Step 5.

Figure 3.12. Flowchart of the proposed partial distortion search with predictive search area

Step 5: Check the value of g . If $g = 19$, update the current SAD_{MIN} with SAD_{19} , move to the next candidate, go to Step 2. Otherwise go to Step 6.

Step 6: Let $g = g + 1$, go to Step 3.

Finally, the candidate MV with minimum SAD will be chosen as the MV for the current MB. Fig. 3.12 shows the flowchart of our proposed algorithm. As can be seen, practically there is no computational overhead for determining the predictive search area since we only make use of the previous neighboring MVs and the computation for median operator is negligible. Besides using median as the predicted motion vector, we have also com-

pared its performance with other prediction methods such as mean and other estimated formula $A + B - C$. From our extensive experiments we observe that the median predictor exhibit slightly better performance than the other two and this will be shown in the next section.

3.3.3 Simulation Results

The performance of the proposed algorithm has been tested using eight standard test video sequences including *Foreman*, *Stefan*, *Coastguard*, *Carphone* and *Miss America* in QCIF format (176×144), and *Mobile Calendar*, *Susie* and *Flower Garden* in CIF format (352×288). Among these video sequences, *Mobile Calendar* and *Coastguard* have 300 frames while others have 100 frames. For the motion characteristics, *Stefan* and *Foreman* have complex motions; *Miss America* and *Susie* have moderate motions while others contain lots of panning and median motions. The MB size is 16×16 and the search window size in the motion estimation is set to $w = 15$. In the simulation, we have tested our proposed algorithm with different PSA sizes ($D=1,2,3$) and compared the performance against the original FS and other partial distortion based fast FS methods including conventional PDS, gradient based PDS (GBPDS) [36], NPDS and APDS. Note that, different from the top-to-bottom and left-to-right scan order in the original PDS, the GBPDS rearranges the accumulation order of pixels when calculating block distortion measure according to their gradient magnitudes. For fair comparison, the prediction of new search window using MV_{pred} is also employed for other partial distortion methods. According to the analysis in [38], the parameter k is initialized to 0.05 in both APDS and our algorithm. In Fig. 3.13 we illustrate the PSNR performance of our proposed algorithm against the quality factor k for sequence *Coastguard*. As can be seen, the PSNR increases

Figure 3.13. PSNR performance of the proposed partial distortion search algorithm against the quality factor k using sequence *Coastguard*.

very fast when k is small and it becomes more stable when k is larger. This characteristic is the same as in the original APDS.

The PSNR degradations of different methods with respect to FS and the probabilities of finding true motion vectors are tabulated in Table 3.6. Generally, the motion estimation accuracy of our method increases when we choose larger PSA sizes. Compared to FS and the first two lossless methods (PDS and GBPDS), the PSNR degradations of our proposed algorithm are no more than 0.07 dB ($D=3$) and our prediction accuracy is very close to the original APDS. On the other hand, without using the adjustable function, the increased erroneous rejections lead to more degradations of the performance of NPDS. Table 3.7 demonstrates the computational complexity comparisons between different approaches. The listed are average operations required per MB which include: absolute, addition, comparison and left-shift. Note that the traditional multiplication operations in NPDS, APDS and our proposed method are translated to the combinations of left-shift and addition. As can be seen, although most accurate, the speed-up ratios of lossless approaches are limited compared with the normalization based PDS. By using smaller partial distortions in the first stage of NPDS, APDS algorithm greatly reduces the

Table 3.6. Comparisons for PSNR and probability of finding true MVs

	Mobile Calendar		Suzie		Flower Garden		Foreman	
	Δ PSNR (dB)	Ave. Prob.	Δ PSNR (dB)	Ave. Prob.	Δ PSNR (dB)	Ave. Prob.	Δ PSNR (dB)	Ave. Prob.
FS	0.00	1.000	0.00	1.000	0.00	1.000	0.00	1.000
PDS	0.00	1.000	0.00	1.000	0.00	1.000	0.00	1.000
GBPDS	0.00	1.000	0.00	1.000	0.00	1.000	0.00	1.000
NPDS	-0.25	0.826	-0.07	0.884	-0.08	0.888	-0.20	0.805
APDS	-0.01	0.946	0.00	0.954	-0.01	0.947	-0.02	0.944
New (D=1)	-0.22	0.893	-0.08	0.881	-0.08	0.897	-0.15	0.911
New (D=2)	-0.14	0.915	-0.04	0.923	-0.05	0.909	-0.10	0.927
New (D=3)	-0.07	0.930	-0.01	0.947	-0.01	0.923	-0.07	0.948
	Stefan		Coastguard		Carphone		Miss America	
	Δ PSNR (dB)	Ave. Prob.	Δ PSNR (dB)	Ave. Prob.	Δ PSNR (dB)	Ave. Prob.	Δ PSNR (dB)	Ave. Prob.
FS	0.00	1.000	0.00	1.000	0.00	1.000	0.00	1.000
PDS	0.00	1.000	0.00	1.000	0.00	1.000	0.00	1.000
GBPDS	0.00	1.000	0.00	1.000	0.00	1.000	0.00	1.000
NPDS	-0.09	0.879	-0.08	0.943	-0.16	0.847	-0.10	0.814
APDS	-0.01	0.961	-0.01	0.990	-0.01	0.944	-0.01	0.931
New (D=1)	-0.14	0.865	-0.08	0.960	-0.16	0.872	-0.09	0.843
New (D=2)	-0.11	0.880	-0.05	0.976	-0.09	0.906	-0.05	0.882
New (D=3)	-0.07	0.928	-0.01	0.987	-0.04	0.929	-0.01	0.904

Table 3.7. Comparisons for average operations per MB and the speed-up gain against FS

	Mobile Calendar		Suzie		Flower Garden		Foreman	
	Operations	Speed-up	Operations	Speed-up	Operations	Speed-up	Operations	Speed-up
FS	667647.0	1.00	667647.0	1.00	667647.0	1.00	600737.0	1.00
PDS	231821.8	2.88	226141.6	2.95	177813.8	3.75	176687.6	3.40
GBPDS	165669.2	4.03	161657.9	4.13	125971.1	5.30	112920.2	5.32
NPDS	44901.0	14.87	45117.9	14.80	46395.3	14.39	39946.9	15.04
APDS	30247.3	20.07	39579.8	16.87	38609.7	17.29	16109.9	37.29
New (D=1)	15415.5	43.31	16264.2	41.05	16416.2	40.67	12594.1	47.70
New (D=2)	15809.8	42.23	16733.0	39.90	16898.2	39.51	12814.4	46.88
New (D=3)	16243.8	41.10	17182.5	38.86	17251.0	38.70	13043.3	46.06
	Stefan		Coastguard		Carphone		Miss America	
	Operations	Speed-up	Operations	Speed-up	Operations	Speed-up	Operations	Speed-up
FS	600737.9	1.00	600737.9	1.00	600737.9	1.00	600737.9	1.00
PDS	141301.3	4.25	136531.1	4.40	118856.6	5.05	214484.6	2.80
GBPDS	109824.1	5.47	104476.0	5.75	87571.1	6.86	144756.1	4.15
NPDS	40285.8	14.91	39534.8	15.20	40257.6	14.92	40649.1	14.78
APDS	20502.0	29.30	15585.9	38.54	18571.2	32.35	34042.7	17.65
New (D=1)	13889.9	43.25	11774.6	51.02	13145.2	45.70	14803.8	40.58
New (D=2)	14293.1	42.03	11907.6	50.45	13382.4	44.89	15348.4	39.14
New (D=3)	14472.0	41.51	12163.1	49.39	13680.1	43.91	15712.2	38.23

computational complexity of NPDS. Furthermore, by introducing predictive search area in our proposed method, we significantly improve the speed-up performance of the original APDS.

Table 3.8. MV predictor performance comparisons for Foreman (D=3)

MV_{pred}	PSNR to FS (dB)	Ave. Probability	Operations	Speedup
Median	0.07	0.948	13043.3	46.06
Mean	0.09	0.930	13096.5	45.87
A+B-C	0.10	0.917	13203.0	45.50

Table 3.9. Average PSNR degradations and speed-up ratios of different algorithms for all sequences

	FS	PDS	GBPDS	NPDS	APDS	Proposed
Δ PSNR(dB)	0.00	0.00	0.00	-0.13	-0.01	-0.04
Speed-up	1.00	3.69	5.13	14.96	26.17	42.22

Table 3.8 demonstrates the simulation results of using different MV_{pred} calculation methods in our proposed algorithm for *Foreman* sequence. Similar results are also observed in other sequences. As can be seen, all three methods provide comparable performance while the median predictor can achieve slightly better results. Table 3.9 shows the average PSNR degradations and speed-up ratios of different algorithms for all sequences. The proposed method achieves on average 42 times speed-up ratio as compared to FS with only 0.04 dB degradation in PSNR. Compared to APDS, our method saves more than 38% computations and maintains very similar PSNR performance. In addition, the proposed algorithm performs much better than other PDS methods in terms of trade-off between computation and quality. Besides, in Fig. 3.14 and Fig. 3.15 we demonstrate frame by frame comparisons of average operations per MB and the probability of finding true MVs for sequences *Mobile Calendar* and *Coastguard*.

Figure 3.14. Frame by frame comparisons of average operations per MB for sequences (a) *Mobile Calendar* and (b) *Coastguard*

Figure 3.15. Frame by frame comparisons of probabilities of finding true MVs for sequences (a) *Mobile Calendar* and (b) *Coastguard*

3.4 Remarks on Fast Search and Fast Matching

It can be seen that, fast block matching motion estimation algorithms are based on either reducing the number of candidates to be examined in the search window (fast search) or early rejections of impossible candidates by calculating partial distortions (fast matching). The success of these two types of fast algorithms depends heavily on their own assumptions. For the first type, they are all implicitly based on the assumption that the block distortion measure increases monotonically as the checking point moves away from the global minimum block distortion point. The underlying assumption for the second type, the partial distortion search based algorithms, is that the total distortions for most candidates are far from the minimum distortion such that many candidates can be rejected only using their partial distortions in an early stage.

Since in the real world video sequences, the above assumptions will not hold for some cases thus the estimation efficiency of the fast block matching algorithms will be affected. Normally, fast search and fast matching algorithms have different strengths and weaknesses. The speed-up gains against FS for fast search algorithms will increase significantly when the search window size is increased while the computation reduction of fast matching based algorithms depends more on the block size. Although the fast search based methods can achieve more computation reductions, their distortions for reconstructed frames are larger than those of fast matching based FS approaches which are nearly lossless. However, the performance of fast matching based algorithms is very sensitive to the size of the blocks in motion estimation. The larger the block size, the higher the speed-up ratios. On the contrary, when the block size is very small the efficiency for partial distortion search will be reduced since the theoretical maximum speed-up gain is in linear relation to the block size chosen. Recently, Zhou *et al* in [113] propose to use an adaptive diversity search strategy by combining the 3SS and DS in

order to take the advantage of both non-center-biased and center-biased methods while overcome the disadvantages of the individual search strategies. In addition, pixel decimation is also been adopted in their scheme. This shows a new direction for motion estimation in which our proposed algorithms are suitable candidates for this combination. It is also observed in [112, 113, 115] that with the greatly increased speed of the CPUs in today's computer, the computation reduction on algorithmic level does not correspond to a similar reduction in runtime. Therefore, to design a dedicated hardware codec is extremely essential for real-time video coding applications [114] and it would be a promising direction to jointly optimize the algorithmic performance and hardware implementations.

3.5 Conclusion

In this chapter, we have presented two fast algorithms for block matching motion estimation. The first algorithm, *i.e.*, an efficient three-step search algorithm (E3SS), belongs to the fast search category in which only part of the candidates in the search window are examined. By employing a small diamond checking pattern in the first step, the unrestricted search step is used to search for the center-biased motion vectors. Experimental results show that for both large and small search windows the proposed E3SS performs better than the 3SS, 4SS, N3SS and DS in terms of MSE with fewer or comparable number of search points for the sequences which contain medium to large motions. The second approach belongs to the fast matching category. We have proposed a novel partial distortion search algorithm which uses a predictive search area for motion estimation. The motivation is to adaptively enhance the prediction accuracy in the predictive search area for the conventional partial distortion methods. Instead of employ-

ing adjustable function in the whole search window as in APDS, we only use it for the area in which the false rejection of candidates occurs most likely. Simulation results show that the proposed algorithm achieves significant computation reduction compared to FS for various kinds of video sequences with only negligible PSNR degradation.

Chapter 4

Smoothness Constrained Block

Matching Criterion

4.1 Introduction

In the previous chapter we have studied fast algorithms for block matching motion estimation. As we know, the basic operation of block matching is to pick up a candidate block in a predefined search range by calculating the matching function between the candidate and the current block. This procedure is repeated until all the possible candidates have been checked and finally the best match is selected.

In this chapter, we focus on the block matching criterion which is the measure of similarity between the current block and the candidate block. The selection of the matching function has a direct impact on the computational complexity and the coding efficiency. We have briefly investigated several traditional block matching criteria in

Chapter 2 including the Normalized Cross-Correlation Function (NCF) [26], Mean Squared Error (MSE) [12], Mean Absolute Difference (MAD) [12], Pel Difference Classification (PDC) [27] and the Minimized Maximum Error (MiniMax) [28]. Among the above criteria, the MAD seems to be the most popular choice in designing practical video coding systems because of its accuracy and simple operations required. However, the MAD criterion is just the average pixel distortion without considering the overall smoothness of the residue block. As a result, when conducting motion estimation using MAD, the block which can minimize the distortion is selected without considering how many bits are required to code the residue. In contrast, the subsequent transform coding tries to code the residue using minimum number of bits. Taking this point into consideration, in this chapter, we propose a novel block matching criterion which aims at eliminating the mismatch between motion estimation and the subsequent transform coding. The proposed criterion is to enhance the conventional MAD scheme with a new smoothness constraint on the residue block. Simulation results show that our proposed criterion performs better than the traditional criteria such as MAD, MSE and MiniMax in terms of PSNR and bit rate of the encoded video.

The rest of this chapter is organized as follows. In Section 4.2, we present the development of the proposed block matching criterion and its implementation. Section 4.3 demonstrates the simulation results for performance comparison. Finally, the conclusion is given in Section 4.4.

4.2 Smooth Constrained MAD Criterion

4.2.1 Motivation

In motion estimation and compensation, the predicted frame is constructed after predictions for all the MBs in the current frame are obtained. Then the residue frame is calculated and coded using the transform coding technique. According to the characteristics of discrete cosine transform (DCT), coding a flat error block which has the same value for all the elements within the block only needs very few bits for the DC coefficient. The remaining AC coefficients should be all zeros thus there is no need for coding them. For those non-flat residue blocks it is clear that the smoother the block is the fewer bits that may be required to code the residue. In view of this fact, a good match between the current block and the candidate block requires not only minimizing the total distortions between the two blocks, the smoothness of the residue block is also highly desirable.

Usually we can calculate the variance of the residue block to determine its smoothness. However, to calculate variance requires so many computations that it is not suitable for real-time implementation. Instead, we use another simple factor as the smoothness measurement, *i.e.*, the difference between the maximum residue and the minimum residue within the residue block, denoted as MMD. Obviously, we can reasonably assume that smoother residue block normally has smaller MMD because all its components will be centralized in a smaller range. In the following we propose a smooth constrained MAD (SCMAD) criterion based on our assumption for choosing the candidate block to improve the coding efficiency of transform coding following the motion estimation.

4.2.2 Development of the SCMAD

In the proposed algorithm, we first subtract the current MB, X (16×16), by the candidate MB, Y (16×16), and form the residue MB, R (16×16) where $r_{i,j} = x_{i,j} - y_{i,j}$ and

Figure 4.1. Four 8×8 residue blocks used in the SCMAD

$i, j = 1 \dots 16$. Then we compute the conventional MAD. After that, we subdivide the residue MB, R into four small blocks (R_1 to R_4) with size 8×8 and determine the MMDs of the four residue R_m separately. This procedure is depicted in Fig. 4.1. As defined before, for R_m , we have

$$MMD_m = r_{\max}^m - r_{\min}^m \quad (4.1)$$

where $r_{\max}^m$ and $r_{\min}^m$ are the maximum residue and the minimum residue within a residue block R_m . The reason behind dividing the 16×16 residue MB into four 8×8 blocks for MMD calculation is as follows. In video compression standards, traditionally, the motion vector is searched on a MB basis while the prediction error or residue is coded using a block (8×8) based DCT transform. Thus, it is better to consider the smoothness of the subdivided residue blocks separately and take the sum of those four MMDs as the overall smoothness measurement for the resulting residue MB. Furthermore, the size of the subdivided residue blocks can be adapted to the type of DCT transform used. For instance, if we choose 4×4 based DCT transform, we may divide the residue MB into 4×4 subblocks accordingly.

Finally, the proposed distortion measure, namely, smooth constrained MAD (SCMAD), is defined as follows:

Figure 4.2. Block diagram of the proposed SCMAD

$$SCMAD = MAD + \alpha \times \sum_{m=1}^4 MMD_m \quad (4.2)$$

where α is a weighting factor and MAD is calculated by

$$MAD = \frac{1}{256} \sum_{i=1}^{16} \sum_{j=1}^{16} |x_{i,j} - y_{i,j}| \quad (4.3)$$

The candidate with the smallest SCMAD is to be chosen as the best match block. The steps for finding the best match block using this new block matching criterion (SCMAD) are summarized below:

- Step 1: Subtract a candidate MB from the current MB and form the residue MB.
- Step 2: Calculate the MAD for the candidate by (4.3).
- Step 3: Subdivide the residue MB into four small residue blocks (R_1 to R_4) and determine the MMD_m based on the residue obtained in Step 1 using (4.1).
- Step 4: Calculate the final distortion measure SCMAD of the candidate using (4.2).
- Step 5: Select the candidate block which has the minimum SCMAD.

The block diagram of the proposed SCMAD is depicted in Fig. 4.2. As can be seen, besides the subtractor, absolute function and the accumulator which are used in the traditional MAD criterion, the additional comparator is required in the proposed SCMAD in order to calculate the MMD . Without using complicated multiplications as in the MSE and NCF criteria, the SCMAD is also quite suitable for both hardware and software implementations. In the following section, we will demonstrate the performance of the proposed block matching criterion and compare it with other traditional criteria.

4.3 Experimental Results

To test the performance of the proposed criterion, four typical video sequences were used in our simulations which include three CIF sequences (*Foreman* 300 frames, *Table Tennis* 100 frames, *Mother & Daughter* 100 frames) and one QCIF sequence (*Miss America* 100 frames). Among these test videos, *Foreman* and *Table Tennis* have relatively fast motions while the other two contain moderate motions. We conducted the simulations based on a H.261 software codec [72]. Three other traditional block matching criteria, *i.e.*, MAD, MSE and MiniMax, were also implemented for performance comparison. In motion estimation the MB size was 16×16 and the block search strategy was full search with a maximum displacement of ± 15 in both directions. A fixed quantization parameter (QP) of 15 was adopted in order not to affect the results by rate control operations. In our experiments, we set the frame rate to 30 fps and the frame coding structure was I frame for the first frame followed by consecutive P frames without frame skipping. In P frames, all MBs were forced to be inter-coded in order to ensure a fair comparison by using the same number of inter-coded MBs for different block matching criteria. Note that, we have introduced an important weighting factor α in (4.2) when

Fig.4.3. R-D performance of SCMAD with different α values. (a) PSNR difference vs. α and (b) Bitrate difference vs. α

calculating the proposed SCMAD. When $\alpha = 0$, the SCMAD becomes the traditional MAD metric and when α increases, the MMD will have more influence on the rate-distortion (R-D) performance. Fig.4.3 illustrates the R-D performance of SCMAD with different α values. As can be seen from this figure, the best value of α is in the range of 6/256 to 8/256 and we empirically chose the α value of 8/256, 6/256, 7/256 and 8/256 for the above four sequences respectively.

The performances of different block matching criteria are tabulated in Table 4.1. We choose two criteria for evaluation which are the average luminance PSNR of the recon-

Table 4.1. PSNR and bit rate comparisons for different block matching criteria

		MAD	MSE	MiniMax	SCMAD
Foreman	PSNR (dB)	29.72	29.70	29.06	29.83
	Bitrate (kbps)	792.80	789.23	908.31	757.13
Table Tennis	PSNR (dB)	27.94	27.94	27.43	27.98
	Bitrate (kbps)	952.10	950.47	1119.156	936.12
Mother & Daughter	PSNR (dB)	31.44	31.47	30.99	31.57
	Bitrate (kbps)	130.69	127.36	161.47	126.31
Miss America	PSNR (dB)	34.69	34.76	34.06	34.79
	Bitrate (kbps)	15.30	15.50	18.89	15.02

structed video and the total number of bits for coding inter-blocks in P frames (residue bits). From Table 4.1 it is clearly shown that by taking the smoothness constraint into consideration, the SCMAD method outperformed the other three criteria for all the four test videos. The proposed SCMAD achieved the best video quality with the smallest number of bits. Generally, we obtained savings in bit rate of around 2-4% over using MAD criterion. Especially for *Foreman*, the number of bits was reduced by 4.5% on average. In addition, by using the new criterion the image quality for all four test videos was slightly improved by 0.04 to 0.13 dB as compared to MAD criterion. As can be seen from Table 4.1, using MSE also improved the performance of MAD while the accuracy of MiniMax criterion was not as good as the other three methods.

Fig. 4.4 shows the coding results of the 67th frame of *Forman* sequence by MAD and the SCMAD criterion. It is observed that by using SCMAD, the number of bits for coding this frame was reduced by 9.3% as compared to the MAD method. Besides, the PSNR of our method was also better than using MAD criterion. In Fig. 4.5 and Fig. 4.6 we have depicted frame by frame comparisons of MAD and SCMAD for *Foreman* and *Mother & Daughter* respectively. As shown in these figures, we achieved consistent improvement in PSNR and bit rate. In other words, our proposed SCMAD can provide better rate-distortion (R-D) performance than the traditional MAD criterion.

(a)

(b)

(c)

Figure 4.4. Performance comparisons for coding the 67th frame of *Foreman* sequence. (a) Original frame (b) Decoded frame by using MAD as block matching criterion. The PSNR is 30.44 and the number of bits is 8258. (c) Decoded frame by using SCMAD as block matching criterion. The PSNR is 30.56 and the number of bits is 7493.

Figure 4.5. Frame by frame comparisons for *Foreman* sequence (a) PSNR (b) Bit rate

(a)

(b)

Figure 4.6. Frame by frame comparisons for *Mother & Daughter* sequence (a) PSNR (b) Bit rate

4.4 Conclusion

In this chapter, we have presented a new motion estimation block matching criterion called smooth constrained MAD (SCMAD) which enhances the conventional MAD function by adding a smoothness control for the residue block. The main advantage of our new scheme over MAD is that it can further distinguish candidate blocks when they have similar MAD values by taking the smoothness constraint MMD into consideration thus leading to a better decision in terms of smaller bit rate. The simulation results have indicated that the SCMAD performs better than MAD and other traditional criteria such as MSE and MiniMax in terms of the R-D tradeoff. The major objective of this work is to provide a new perspective to block motion estimation. Although visually, the R-D performance improvement is not very significant, the experimental results show that the SCMAD has the potential to improve the performance of block matching motion estimation. In addition to the heuristic method, an automatic scheme to determine the weighting factor α may be desirable and it would be considered for further exploration in future works.

Chapter 5

Efficient Inter Mode Decision for H.264/AVC Video Coding

5.1 Introduction

The traditional fixed size block matching (FSBM) motion compensation is based on the assumption that in a video frame, each block of pixels will undergo the same translational motion. However, this assumption will not always hold since in the real-world video sequence complex motion contents may exist within each block. Thus, it is not possible to represent the complex motion using only one motion vector. As shown in Chapter 2, one solution is to employ variable size block matching (VSBM) strategies. VSBM based motion compensation is a very important technique for video coding.

The latest H.264/AVC video coding standard [76] achieves much higher compression gain than the previous standards. This is mainly due to the fact that the H.264/AVC

standard adopts more complicated approaches in the coding procedure. One of these advanced techniques is the use of VSBM in the rate-distortion optimized (RDO) [31, 32] mode decision. Totally seven different block types are enabled in the motion compensation. Using VSBM can significantly improve the coding performance compared with the previous video coding standards. On the other hand, the computational complexity of H.264/AVC encoder increases dramatically due to the various block types used. In this chapter, we propose an efficient inter mode decision approach for H.264/AVC video coding. The objective is to reduce the number of candidate block types in the RDO based motion compensation while maintaining the coding efficiency of the original scheme. Our method is based on the macroblock (MB) classification which can effectively classify MBs into two categories, *i.e.*, homogeneous or non-homogeneous. The basic idea is to save computations of the RDO based mode decision for homogeneous MBs by avoiding the use of small sized block types.

The rest of this chapter is organized as follows. In Section 5.2, we will review the standard mode selection scheme in H.264/AVC video coding. In Section 5.3, our efficient block size selection method for inter mode decision is presented. The simulation results of our proposed algorithm are given in Section 5.4. Finally, the conclusion is drawn in Section 5.5.

5.2 Standard Mode Selection Scheme in H.264/AVC

In H.264/AVC coding standard, in addition to the block sizes of 16×16 and 8×8 which are allowed in the previous standards, other block sizes are also enabled for inter frame

Figure 5.1. Different block types used in H.264/AVC inter frame coding

coding including 16×8 , 8×16 , 8×4 , 4×8 and 4×4 . These seven block types are depicted in Fig. 5.1. In inter frame coding, the possible modes for a MB are as follows [55]:

$$\text{mode} \in \left\{ \begin{array}{l} \text{SKIP, Intra-}4 \times 4, \text{ Intra-}16 \times 16, \\ \text{Inter-}16 \times 16, \text{ Inter-}16 \times 8, \text{ Inter-}8 \times 16, \text{ Inter-}8 \times 8 \end{array} \right\} \quad (5.1)$$

where Inter- 8×8 mode consists of four sub-modes: Inter- 8×8 , Inter- 8×4 , Inter- 4×8 and Inter- 4×4 . Note that, the SKIP mode represents the direct copy from the reference frame. Intra- 4×4 and Intra- 16×16 are intra modes predicted from the adjacent boundary pixel values of the previous encoded MBs. In order to choose the best encoding mode for the current MB, the H.264/AVC standard will exhaustively examine all possible modes and select one with the minimum Lagrangian RDO cost which is given by:

$$J(s, c, \text{MODE} | \text{QP}, \lambda_{\text{MODE}}) = \text{SSD}(s, c, \text{MODE} | \text{QP}) + \lambda_{\text{MODE}} \cdot R(s, c, \text{MODE} | \text{QP}) \quad (5.2)$$

$$\lambda_{\text{MODE}} = 0.85 \times 2^{(\text{QP}-12)/3} \quad (5.3)$$

where QP is the quantization parameter for this MB, λ_{MODE} is the Lagrange multiplier for mode decision, SSD is the sum of the squared differences between the original block s and its reconstruction c and MODE is one of the potential prediction modes. The rate R is given as the sum of the bits for the mode information, the motion vectors, as well as the

Figure 5.2. R-D performance comparisons between VSBM and FSBM using H.264/AVC codec

transform coefficients. Different from the previous standards, by utilizing more block types in H.264/AVC inter frame coding, more accurate motion compensation can be achieved for those blocks which contain complex motions. As a result, the rate-distortion (R-D) performance of the traditional schemes which are based on FSBM can be greatly improved. This improvement can be seen from Fig. 5.2. In Fig. 5.2, the FSBM represents that only 16×16 block size is used in motion compensation whereas all seven block types are enabled in the VSBM scheme.

Although this R-D performance improvement is highly desirable for a video coding standard, the increase of the block types used makes the computational burden of H.264/AVC encoding process far more demanding than all of the previous video coding standards. In the following, we will introduce our proposed fast inter mode decision method which can provide better tradeoffs between complexity and quality.

5.3 Efficient Inter Mode Decision Algorithm

Generally, small sized blocks are suitable for MBs which belong to the non-homogeneous regions in a video frame such as the blocks at the boundary between the moving object and the static background. However, exhaustively examining all possible combinations of different block types for those MBs in the homogeneous regions which contain consistent motion will not provide any improvement than using only large block types. Therefore, it is not necessary to examine smaller sized block types for these homogeneous MBs. As a result, computations can be saved due to the fewer block types used in the RDO based inter mode decision. This concept is the basic motivation of our proposed inter mode decision scheme.

5.3.1 Macroblock Classification

Previously, Huang *et al.* [51] classified the MBs into background, shade motion and edge motion blocks by a two-stage classifier. In their method, motion vectors for background blocks were directly set to zero. For shade motion blocks, pixel subsampling technique was used to reduce the computation of motion estimation. The variable size block matching was only conducted for edge motion blocks. In order to classify MBs as homogeneous or non-homogeneous, a pre-computed edge map with Sobel operators was used in [50]. If the amplitude of the edge vectors within a block is less than a threshold, it will be classified as homogeneous. Furthermore, inter modes of smaller sized blocks will be disabled for homogeneous blocks.

Compared with the multi-stage classifier and the edge map based methods, our proposed fast inter mode decision algorithm utilizes a more simple and direct measurement to classify each MB in a frame, *i.e.*, the absolute difference between consecutive frames.

Figure 5.3. Relations between frame differences and the block types used in H.264/AVC (a) Original frame from the sequence *Mother & Daughter* (b) Difference frame between the original frame and the previous frame (c) Decoded frame with corresponding macroblock types by the H.264/AVC encoder

In general, the frame difference contains lots of information about the motion contents in successive frames. Large amplitudes will appear on the moving edges or boundaries of moving objects while small amplitudes in homogeneous areas. So if the amplitudes in a MB are small, it is most likely that this MB belongs to a homogeneous region and using only larger block sizes in motion estimation will be accurate. Otherwise, this MB may contain complex motions and using more block types can achieve better rate-distortion performance. Fig. 5.3 shows an example of the relations between frame differences and the resulted block types after the RDO mode decision. As expected, the areas in Fig. 5.3(c) with small sized blocks are well matched with the regions in Fig. 5.3(b) which have relatively large amplitudes. In the proposed fast inter mode decision algorithm, the mean absolute frame difference (MAFD) of the current frame and mean absolute difference

(MAD) of the current MB are used to determine whether the current MB belongs to homogeneous regions or not. They can be calculated using (5.4) and (5.5) respectively:

$$MAFD = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N |x_{i,j} - y_{i,j}| \quad (5.4)$$

$$MAD = \frac{1}{256} \sum_{i=1}^{16} \sum_{j=1}^{16} |x_{i,j} - y_{i,j}| \quad (5.5)$$

where $x_{i,j}$ and $y_{i,j}$ denote the intensity levels of pixels at location (i, j) of the current frame and its previous frame and M, N are the horizontal and vertical dimensions of the frame. If the MAD satisfies the following condition (5.6),

$$MAD \cdot w \leq MAFD \quad (5.6)$$

this MB is considered as in a homogeneous region and only large block types will be used in its RDO mode decision. Otherwise it belongs to a non-homogeneous region which may contain complex motions or having moving edges thus the rest smaller sized block types will also be enabled. Note that, there is a weighting factor w in (5.6). The determination of this weighting factor depends on the QP for the MB. Normally smaller QP will result in using more small block types while larger QP will lead to use of more large block types. Therefore w is used in order to add a bias for using small block types when the QP is small and vice versa.

5.3.2 The Algorithm for Inter Mode Decision

The complete algorithm for inter mode decision is summarized as follows:

- Step 1: Before encoding the first MB of the current frame, subtract the current frame by its previous frame and calculate the MAFD using (5.4)
- Step 2: For the current MB, calculate the MAD value using (5.5). Conduct MB classification by examining the inequality (5.6). If it is satisfied, then this

Figure 5.4. Flowchart of the proposed inter mode decision algorithm

MB belongs to a homogeneous region, go to Step 3. Otherwise this MB belongs to a non-homogeneous region, go to Step 4.

Step 3: Conduct rate distortion optimized (RDO) motion estimation and mode decision using block types 16×16 , 16×8 and 8×16 . Go to Step 5.

Step 4: Conduct rate distortion optimized (RDO) motion estimation and mode decision using all seven block types 16×16 , 16×8 , 8×16 , 8×8 , 8×4 , 4×8 and 4×4 . Go to Step 5.

Step 5: Choose the best coding mode for the current MB by comparing the result from Step 3 or Step 4 with the results of SKIP mode and Intra mode. Conduct the actual encoding process for this MB.

Fig. 5.4 illustrate the flowchart of our proposed efficient inter mode decision algorithm. For simplicity, we have adopted the w values 1.2, 1.0, 0.8, 0.6 for QP=28, 32, 36, 40 respectively in this work. Furthermore, in calculating (5.4) and (5.5) the previous original frame is used rather than the previous reconstructed frame. This is because coding distortions in the reconstructed frame may affect the accuracy of the MB classifier. By employing such classification algorithm, the computational complexity of the original encoder can be greatly reduced while the coding efficiency is maintained. The performance of our proposed mode decision algorithm will be illustrated in the next section.

5.4 Experimental Results and Discussions

The proposed mode decision approach was tested using six standard testing video sequences including four QCIF sequences with 100 frames (*Foreman*, *Carphone*, *Miss America*, *Mother & Daughter*) and two CIF sequences with 300 frames (*Table Tennis* and *Paris*). In these test videos, *Foreman*, *Table Tennis* and *Carphone* have relatively large motions while the other three have moderate motions. The experiment was carried out using the JVT JM11 reference encoder [101]. In the motion estimation, five reference frames were enabled with the maximum search range of 32 and the motion vector resolution was 1/4 pixel. Context adaptive binary arithmetic coding (CABAC) was adopted as the entropy coding method. The Hadamard transform was enabled. The frame rate was 30 fps and the frame coding structure was IBBP. Experiments were conducted for four quantization parameters QP=28, 32, 36 and 40.

Table 5.1. PSNR and bit rate comparisons between the full mode search and the proposed inter mode decision algorithms

	Foreman QCIF				Carphone QCIF			
	QP=28	QP=32	QP=36	QP=40	QP=28	QP=32	QP=36	QP=40
Δ PSNR(dB)	-0.05	-0.05	-0.01	-0.04	-0.04	-0.01	-0.01	0.01
Δ Bitrate(%)	0.2	0.3	-0.1	-1.3	0.5	0.4	-0.4	-1.3
	Miss America QCIF				Mother & Daughter QCIF			
	QP=28	QP=32	QP=36	QP=40	QP=28	QP=32	QP=36	QP=40
Δ PSNR(dB)	-0.03	-0.01	-0.03	-0.03	-0.05	-0.01	-0.03	-0.04
Δ Bitrate(%)	-0.1	-1.1	0.3	-0.7	0.3	-0.3	-0.5	-0.8
	Table Tennis CIF				Paris CIF			
	QP=28	QP=32	QP=36	QP=40	QP=28	QP=32	QP=36	QP=40
Δ PSNR(dB)	-0.01	-0.03	-0.03	-0.03	-0.08	-0.07	-0.04	-0.03
Δ Bitrate(%)	0.8	-0.5	-0.9	-0.8	1.3	1.0	0.1	-0.1

Table 5.2. Computational complexity comparisons between the proposed inter mode decision algorithm and the full mode search scheme

	Average No. of block types	Δ Complexity(%)
Foreman	5.00	-29
Carphone	4.36	-38
Miss America	3.64	-48
Mother & Daughter	4.10	-41
Table Tennis	5.39	-23
Paris	4.62	-34

The PSNR and bit rate comparisons between the original encoder and the one with our proposed inter mode decision method are tabulated in Table 5.1. Moreover, in Fig. 5.5 we show the complete R-D curves of both algorithms for all the test sequences. As can be seen, the coding results for both methods are very similar. Compared with the original encoder which utilizes the exhaustive search method in mode decision, the PSNR degradations of the proposed fast inter mode decision algorithm are no more than 0.08 dB. Moreover, in many cases bit rate savings are achieved. On average, the proposed scheme achieved 0.2% bit rate saving at the cost of 0.04 dB PSNR degradation. In other words, the R-D performances of both algorithms are almost the same. The computational complexity comparisons are illustrated in Table 5.2. Instead of examining all 7 block

Table 5.3. Performance comparison of different mode decision algorithms

	Proposed method			Grecos's method			Proposed + Greco's		
	Δ PSNR (dB)	Δ BR (%)	Δ T (%)	Δ PSNR (dB)	Δ BR (%)	Δ T (%)	Δ PSNR (dB)	Δ BR (%)	Δ T (%)
TableTennis	-0.02	-0.50	-23.32	-0.03	0.11	-50.61	-0.04	0.48	-58.37
Paris	-0.04	0.70	-31.00	-0.04	1.04	-54.43	-0.07	1.50	-61.04
Foreman	-0.04	-0.23	-30.08	-0.05	0.35	-61.54	-0.09	0.63	-66.52
M&D	-0.03	-0.33	-40.02	-0.05	0.37	-61.14	-0.06	0.94	-65.81
Carphone	-0.02	-0.20	-35.14	-0.03	0.42	-60.67	-0.08	0.81	-67.44
MissAmerica	-0.02	-0.40	-45.68	-0.08	0.55	-64.25	-0.11	0.78	-70.49

types in RDO mode decision for every MB, only 4 to 5 block types were actually utilized in our scheme. On average we have achieved equivalent complexity saving in inter mode decision of 35% compared with the original method. Fig. 5.6 shows two examples of the MB classification results and the final mode decisions for *Foreman* and *Mother & Daughter* sequences respectively.

With the development of fast inter mode decision techniques, more conditions are involved to reject the unlikely candidate modes in an earlier stage thus more speedup gains can be achieved. Recently, Grecos *et al* in [118] present an efficient fast mode decision algorithm for H.264. In Grecos's method, a combination of the conditions for SKIP mode decision, two heuristics to reduce the number of candidate block types and the monotonicity of cost function is adopted. As a result, complexity of mode decision can be reduced significantly. Besides the original full-mode search algorithm, Grecos's method is also selected as a benchmark for comparisons. The experimental results of Grecos's method and our proposed method are tabulated in Table 5.3. Note that, three metrics Δ PSNR, Δ Bitrate and Δ Time are the average values for all QPs as compared to the original full-mode search algorithm. As can be seen from this table, an extra speedup gain (20%~30%) can be achieved in Grecos's method at the cost of slightly degraded R-D performance. This is due to the fact that more heuristics and criteria have been adopted to

Figure 5.5. R-D performances of both algorithms for (a) QCIF sequences and (b) CIF sequences

classify the MB's characteristics. Compared to our simple classification criterion based on MAD value, Grecos's scheme can reject more improper modes in advance. In addition, if the two methods are combined together, more significant speedup gain (up to 70%) can be observed as can be seen from Table 5.3. Although our algorithm now is not as computationally efficient as the latest method, it can have the potential to be used in conjunction with other latest fast mode decision algorithms [116,117,119] to further improve the mode decision efficiency.

Figure 5.6. Examples of the MB classification results and the final mode decisions for sequences *Foreman* and *Mother & Daughter*. (a) Original frame (b) MB classification results. The MBs in white are classified as non-homogeneous. (c) Decoded frame with corresponding MB modes by the reference encoder with full mode search scheme. (d) Decoded frame with corresponding MB modes by the proposed method

5.5 Conclusion

In this chapter, an efficient inter mode decision algorithm for H.264/AVC standard has been proposed. The motivation is to predict the potential block types before conducting the RDO based motion estimation. A classifier has been employed to decide which block types are to be enabled for each MB. It is based on the observation that non-homogeneous regions in a frame normally have large frame differences while homogeneous regions have small frame differences. Experimental results show that compared with the traditional exhaustive search based mode decision approach, the proposed method achieves significant computation savings by up to 48% while maintaining the same rate-distortion performance. Currently, the values of w in our fast mode decision algorithm are determined empirically from our experiments and it is also desirable to have an automatic scheme to determine or update w in the future studies.

Chapter 6

Rate Control for H.264/AVC with Frame Complexity Estimation

6.1 Introduction

As stated in Chapter 2, rate control techniques have been widely used in compressed video communications. The main objective of rate control is to regulate the output bit rates of video encoders to achieve the best utilization of the available channel bandwidth. Benchmark rate control schemes have been developed for various video coding standards such as MPEG-2 TM5 [73], H.263 TMN8 [74] and MPEG-4 VM18 [75]. These schemes, although with different computational complexities, work on the same principle of rate control. They all depend on predefined rate-quantization (R-Q) models. By using these models the quantization parameter (QP) for the current coding unit can be calculated according to the target bit budget that is allocated to this unit. The coding unit can be a

frame, a MB, a video object or a group of consecutive MBs. QP is the most important parameter which directly determines the output bit rate of the video encoder. In addition, different from former standards, QP is also a crucial factor in the rate-distortion optimization (RDO) of the latest H.264/AVC video coding standard [76].

In this chapter, we address the problems associated with frame-level rate control for real-time H.264/AVC video coding. We develop two frame complexity based schemes to enhance the existing H.264/AVC rate control method adopted by Joint Video Team (JVT). By investigating the characteristics of current frame, we first construct an efficient intra-frame R-Q model which can be adaptable to the change of frame complexities. Furthermore, we introduce an improved frame MAD prediction model. Based on the refined MAD, a new frame level bit allocation method is obtained. Generally speaking, more bits will be allocated to the frames with higher complexity and vice versa. Experimental results reveal that by applying our proposed schemes, smaller visual quality variation and improved overall quality of the coded sequences can be achieved. In addition, the frame bits mismatch for both intra- and inter-frames is reduced.

The rest of this chapter is organized as follows. In Section 6.2, we will present the background of H.264/AVC rate control and introduce some related work. In Section 6.3, we describe in detail the development of our proposed intra-frame R-Q model. The enhanced MAD prediction model and frame level bit allocation scheme are introduced in Section 6.4. Section 6.5 demonstrates the experimental results for comparison. Finally, we draw a conclusion in Section 6.6.

6.2 Background and Related Work

H.264/AVC is so far the most efficient standard for video compression which achieves much higher coding efficiency than previous ones. This is because that H.264/AVC employs more advanced approaches in the coding procedure. As a result, the rate control scheme for this new standard also has its new characteristics which are different from the previous standards.

6.2.1 Characteristics of H.264/AVC Rate Control

One of the important approaches is the Lagrangian-function-based RDO in motion estimation and mode decision. The best coding mode for each MB is selected by choosing the one which can minimize the overall cost function $J = D + \lambda \cdot R$ where D and R are the mean square error and the number of bits for coding this MB. As QP is required when calculating the Lagrangian multiplier λ for the current MB before it is actually encoded, rate control for H.264/AVC is facing a new challenge. In order to perform RDO, QP for the current MB must be known. However, when using R-Q model to determine QP, the actual statistics such as the mean absolute difference (MAD) or the residue variance are indispensable which are only available after the RDO process. This is a typical *chicken and egg* dilemma [77]. One possible solution to this kind of dilemma may be using multi-pass based schemes [79, 84, 95]. Although the rate control accuracy can be gradually

Figure 6.1. *Chicken and egg* dilemma in H.264/AVC rate control

improved through QP refinement, due to their high computational complexity they are not suitable for real-time low-delay encoding. Li *et al.* in JVT-G012 [77] proposed an adaptive rate control framework for H.264/AVC and it has been adopted by the JVT. This single-pass rate control method is based on the classic quadratic R-Q model [81, 82, 96] and a linear model for MAD prediction is employed to solve the above dilemma. Using frame level rate control as an example, firstly, the target bit budget for the current P-frame is determined by the average available bit budget and further modified according to the buffer status. Then the MAD of the previous P-frame is used to predict the MAD for current frame. Finally, substituting this predicted MAD for the actual MAD in the quadratic R-Q model, the QP of current frame is thus obtained.

Basically, the adaptive rate control in [77, 78] has an implicit assumption that the video source is stationary which is not always true for real-world video sequences. Based on this assumption, in the frame level bit allocation stage, only buffer status and available channel bandwidth are considered regardless of the characteristics of the current frame. In other words, frames with different coding complexities may have the same bit budget. As a result, large quality fluctuations will occur among the reconstructed video frames. Similarly, without considering the feature of current frame, the performance of the linear MAD prediction model degrades significantly when there is a large change in motion or scene change occurs. This will seriously affect the accuracy of the quadratic R-Q model resulting in large bit rate mismatch. Consequently, the buffer may be filled up quickly and undesirable frame skipping will occur which leads to the motion jerkiness. Moreover, in the adaptive rate control scheme, no explicit R-Q model for intra-frame is discussed. They propose to compute the QP for intra-frame based on the average QP of P-frames in the previous group of pictures (GOP). And this QP is adaptive to both the GOP length and the available channel bandwidth. However, given the bit budget and buffer constraint

when encoding the current intra-frame, it is difficult to accurately estimate the QP according to their scheme. Thus an efficient R-Q model for intra-frames is highly desirable for H.264/AVC which must be both accurate and frame-complexity adaptive.

6.2.2 Related Work for Rate Control

In Xie and Zeng's work [85, 86], to follow the non-stationary characteristics of video scenes, a sequence-based rate control framework is proposed. The actual MAD of current frame is used as the complexity measure and the frame bit allocation will be adjusted according to its complexity. Extending this work to H.264/AVC, Jiang *et al.* [91, 93] and Xu *et al.* [92] predict the frame complexity based on the estimated MAD using linear prediction model. Since this linear model remains the same, incorrectly predicted MAD will affect the frame level bit allocation. Pan *et al.* in [97] propose an adaptive intra-frame R-Q model suitable for low-bit-rate coding. Because it needs actual DCT statistics of current frame before calculating the QP, it is not appropriate to be applied directly to H.264/AVC. Kamaci *et al.* [98] have introduced a Cauchy-Density-Based rate model for H.264/AVC which is more accurate for intra-frames. However, with a simple scheme for updating model parameters, there is not sufficient adaptability to the changes of frame complexity. He *et al.* [83, 84] proposed a linear ρ -domain source model and applied this model to H.264/AVC using a two-loop structure. This method also belongs to the multi-pass category and it is more appropriate for offline coding.

In this work, we are focusing on the single-pass rate control for real-time encoding. The success of a single-pass-based rate control algorithm depends heavily on the R-Q models used in the QP decision stage. In the following sections, we will focus on the development of our proposed frame complexity based R-Q models.

6.3 Intra-frame R-Q Modeling

The intra-coding algorithm of H.264/AVC differs from previous video coding standards in that intra prediction in spatial domain is conducted. For each MB, there are two intra prediction types *i.e.*, intra 16×16 and intra 4×4 . For both types, each 16×16 or 4×4 block is predicted from its spatially adjacent pixels according to the particular prediction mode [76]. Finally, the best intra prediction mode for this MB is determined by which the overall cost is minimized. By utilizing this spatial prediction, the compression performance for intra-frames has been greatly improved over prior schemes.

6.3.1 Traditional Method in JVT

In the adaptive rate control for H.264/AVC [77], the initial QP of the i th GOP is predefined based on the available channel bandwidth and the GOP length. Normally, a smaller QP will be chosen when the channel bandwidth is high and a larger QP is more suitable if the channel bandwidth is low. More specifically, the QP of the current intra-frame in the i th GOP, $QP_i(I)$ is calculated by

$$QP_i(I) = \max \left\{ \min \left\{ \overline{QP}_{i-1}(P) - \min \{2, N_{i-1}/15\}, QP_{i-1}(I) + 2 \right\}, QP_{i-1}(I) - 2 \right\} \quad (6.1)$$

where $\overline{QP}_{i-1}(P)$ is the average QP of P-frames in the previous GOP and N_{i-1} is the length of the previous GOP. Note that, although equation (6.1) can determine the QP according to channel condition and GOP length, it has not provided a complete solution to the intra rate estimation problem. In other words, it does not explicitly define the intra-frame R-Q model. So the potential problem of this scheme is that when the characteristics of current intra-frame change greatly from previous intra-frames, the QP obtained from (6.1) may no longer be suitable. In our scheme, we aim to propose a frame complexity based adaptive R-Q model for intra-frame coding.

Figure 6.2. Intra bit rate of *Foreman* sequence under different QPs

6.3.2 Gradient-based Complexity Measure for Intra-frames

We have exhaustively encoded various test sequences using only intra-coding scheme in H.264/AVC with different QPs. The actual output bit rate of each frame is collected for every QP used. For example, Fig. 6.2 shows the bit rate results for *Foreman* sequence. As can be seen from this figure, for the same quantization value, encoding different frames will consume different numbers of bits. This difference varies with the changes of QPs. The smaller the QP, the greater differences between frames will be observed. In addition, we can find that the shapes of these six curves are highly correlated. In other words, regardless of the quantization value, the output bit rate of each intra-coded frame is inherently determined by its frame characteristics.

There are many activity measures or complexity measures for still image coding. In [99], Kim *et al.* have classified these measures into four categories and suggested that the

Figure 6.3. Gradient per pixel and average pixel value for different frames of *Foreman*

gradient-based method is more reliable for still image coding. In our proposed intra R-Q model, we define the complexity measure of intra-frame based on the average gradient per pixel of the frame. The calculation of gradient is defined by

$$G = \frac{1}{M \times N} \left\{ \sum_{i=1}^{M-1} \sum_{j=1}^{N-1} \left(|I_{i,j} - I_{i+1,j}| + |I_{i,j} - I_{i,j+1}| \right) \right\} \quad (6.2)$$

where M and N are the horizontal and vertical dimensions of the frame respectively. $I_{i,j}$ denotes the luminance value of pixel at the location of (i, j) . Fig. 6.3 illustrates the average gradient value and the average pixel value for *Foreman* sequence. By comparing Fig. 6.3 with Fig. 6.2, we can make another observation that the gradient value can better represent the complexity of an intra-frame since the shapes of these two sets of curves are quite similar. For the same QP, the larger the gradient value the larger the number of bits of this frame.

Figure 6.4. Scatter plots of bit rate versus gradient value for *Foreman* sequence. Corresponding QP values (from top to bottom, left to right) are: 24, 28, 32 and 36.

To further illustrate the relationship between the gradient and the output bit rate, we show a set of scatter plots of bit rate versus gradient value in Fig. 6.4. Obviously, there exists a linear relationship between these two factors. Moreover, with the QP becoming larger, the slope of each “line” is getting smaller and the linear relationship is not as clear at very low bit-rate as at high bit-rate. This may due to the influence of the larger portion of header bits when the bit rate is low. From this linear correlation between the number of bits and the gradient we can draw a conclusion that for a fixed QP, the output bit rate of one intra-coded frame is proportional to its average gradient. Besides this linear relationship, there are another two advantages of using gradient as the intra-frame complexity measure. One is that the computation for calculating gradient is low and the other is the gradient statistics can be obtained from the original frame without any encoding process. These properties are highly desirable for H.264/AVC intra-frame rate estimation.

6.3.3 Adaptive Intra-frame R-Q Model

In the quantization scheme of the previous video coding standards, the quantization parameter QP is directly used as a scaling factor which controls the coding bit rate and the picture quality. For example, in H.263 the quantization step size or scaling factor actually used has a linear relation with QP, *i.e.*, $Q_{step} = 2QP$. This linear relation no longer exists in the latest H.264 standard. Actually, there are 52 values of Q_{step} used in H.264/AVC which are indexed by 52 QPs ranging from 0 to 51 [76]

$$Q_{step} = 2^{\text{floor}(QP/6)} \cdot f(QP\%6) \quad (6.3)$$

$$f(0) = 0.625; f(1) = 0.6875; f(2) = 0.8125; f(3) = 0.875; f(4) = 1; f(5) = 1.125 \quad (6.4)$$

As can be seen from (6.3) and (6.4), the value of Q_{step} doubles for every increment of 6 in QP. Theoretically, rate is more directly related with Q_{step} . Although QP monotonically increases as Q_{step} , we model the output bit rate as a function of Q_{step} in our proposed scheme.

Experimental observations have shown that the intra-frame bit rate can be formulated by

$$R(Q_{step}) = G \cdot f(Q_{step}) \quad (6.5)$$

$$f(Q_{step}) = a \cdot Q_{step}^b \quad (6.6)$$

where G is the average gradient per pixel calculated from (6.2), and $a > 0$, $b < 0$ are parameters that depend on the picture content. In [98] Kamaci *et al.* have empirically justified that the Cauchy density performs better than the traditional Laplacian density for estimating the distribution of the DCT coefficients. The rate estimation accuracy is greatly improved when using the Cauchy-based entropy function especially for intra-coded frames. Furthermore, they have suggested using a simplified equation as (6.6) to approximate the Cauchy entropy function. Based on our observations, we also adopt (6.6)

Figure 6.5. Curve fitting result for the first frame of *Suzie*. The gradient value $G=9.74$ and the model parameters in $f(Q_{step}) = a \cdot Q_{step}^b$ are $a = 0.638$, $b = -0.795$

as a part of our R-Q model (6.5) with the combination of the gradient value G which is an important scaling factor. In other words, our R-Q model can be explained in this way: the bit rate of one intra-frame is proportional to its gradient value and the slope of this linear relation can be represented by a power function of the quantization step size Q_{step} . Fig. 6.5 gives an example of curve fitting result for the first frame of *Suzie* sequence. It shows that the proposed R-Q model achieves accurate estimation of the actual bit rate. However, we also find that it is unlikely to have one single model with fixed parameters to be used for all kinds of frames. Ideally, these two parameters (*i.e.*, a and b) can be precisely determined by solving a set of equations based on the Cauchy density parameter [98]. Because we do not have access to the actual distribution of DCT coefficients in advance,

we need to estimate the parameters based on previous coding results. In the following, we propose an efficient procedure for adaptively updating our R-Q model parameters.

Based on our extensive experiments, we have found that the typical values of b are in the range of -0.90 to -0.70. For simplicity, we choose a moderate value of $b = -0.80$ in our R-Q model. During our parameter updating procedure, the value of b is fixed and we only update parameter a using

$$a_{k+1} = \alpha \times a_k + (1 - \alpha) \cdot \frac{R_k}{G_k \cdot (Q_{step})_k^b} \quad (6.7)$$

where R_k , G_k and $(Q_{step})_k$ are the actual output bit rate (in bits per pixel), the average gradient value per pixel and the real quantization step size for the k th intra-frame. After coding the first intra-frame using predefined $(Q_{step})_0$, a_0 can be calculated as

$$a_0 = \frac{R_0}{G_0 \cdot (Q_{step})_0^b} \quad (6.8)$$

Note that, when $k \geq 1$, after coding the current frame k , there are two reference values that can be used for predicting a_{k+1} . One is the previous predicted a_k which has just been used for calculating $(Q_{step})_k$ and the other one is the actual value of a_k which is represented by $R_k / G_k \cdot (Q_{step})_k^b$. Eq. (6.7) takes into consideration of both values by introducing a forgetting factor α . Normally, when α approaches 1, the predicted value a for the following frame will remain to be a_0 . On the other hand, if α approaches 0, the predicted value a for the following frame will just be set to the actual value of a for the previous I-frame. Similar to [98], a fixed value of $\alpha = 0.5$ is adopted in our scheme. Comparing our proposed R-Q model with the Cauchy-based model, the major difference is that we have considered the gradient based frame complexity measure as an important scaling factor in bit rate estimation. As a result, our model is more adaptive to the changes of frame characteristics.

Table 6.1. Comparison of intra rate estimation performance

Sequence	$QP_0 = 26$			$QP_0 = 32$			$QP_0 = 38$		
	Cauchy	Proposed	Reduction (%)	Cauchy	Proposed	Reduction (%)	Cauchy	Proposed	Reduction (%)
Foreman	2.69	2.24	16.7	3.48	2.98	14.4	3.31	2.95	10.9
Stefan	2.15	1.92	10.7	2.22	1.99	10.4	1.65	1.51	8.5
Trevor	4.73	3.66	22.6	4.38	3.55	18.9	3.63	3.09	14.9
Carphone	2.31	2.14	7.4	2.60	2.43	6.5	3.61	3.45	4.4
Suzie	3.14	2.61	16.9	3.44	2.96	14.0	3.69	3.30	10.6
News	2.29	1.90	17.0	2.40	2.02	15.8	1.91	1.70	11.0
Football	10.81	3.69	65.9	10.77	4.80	55.4	11.37	5.29	53.5
Table Tennis	7.73	3.68	52.4	6.27	4.38	30.1	5.32	4.49	15.6
Coastguard	4.05	3.04	24.9	3.80	2.87	24.5	5.83	5.05	13.4
Comb1	25.41	6.18	75.7	27.49	7.09	74.2	28.69	9.62	66.5

6.3.4 Model Performance Evaluation

To evaluate the performance of our proposed intra-frame R-Q model, we have applied our scheme to a H.264/AVC reference encoder JM11 [101] and encoded a number of video sequences only using intra-coding mode. The experiments were conducted using the first 150 frames of six standard QCIF sequences (including *Foreman*, *Stefan*, *Trevor*, *Carphone*, *Suzie* and *News*) and three CIF sequences (including *Football*, *Table Tennis* and *Coastguard*). In addition, we synthetically created a new test sequence *Comb1* by cascading 30 frames from the six QCIF test videos (five frames from each sequence which are spaced 30 frames apart). The purpose of using *Comb1* sequence is to test the adaptability of the proposed algorithm. We also implemented Cauchy-based method [98] to the encoder for comparison. Since the original scheme for intra-frame QP determination in (6.1) is not suitable for bit rate estimation, it was not included in our comparisons. To evaluate the accuracy of each model, the frame bit-rate-mismatch ratio defined in [98] is adopted as a measure which is calculated as follows:

$$M = \frac{|R_{target} - R_{actual}|}{R_{target}} \times 100\% \quad (6.9)$$

Figure 6.6. Intra-frame bit rate of different rate control schemes for *Comb1* sequence

where R_{target} is set as the actual number of bits of the first frame which will be coded using a predefined QP. (*i.e.*, QP=32 was used in our simulations). Table 6.1 tabulates the average value of M for different sequences with various initial QPs ($QP_0=26, 32$ and 38). From Table 6.1 we can see that the rate estimation accuracy has been greatly improved especially for the combined sequence. On average, the mismatch of our model is about 3.5% and we have achieved up to 75% improvement over the traditional Cauchy-density-based method. Fig. 6.6 and Fig. 6.7 demonstrate frame by frame comparisons for the combined sequence. As can be seen from these figures, our model is more adaptive to frame content changes. This is due to the fact that our scheme can efficiently adjust the model parameter based on the frame complexity measure. Compared to Cauchy-density-based model and the one without rate control, no abrupt bit rate fluctuation is observed in our scheme. Therefore, our R-Q model is more robust and efficient for H.264/AVC intraframe rate control.

Figure 6.7. Intra-frame QP determined by different rate control schemes for *Comb1* sequence

6.4 Frame MAD Prediction and Frame Level Bit Allocation

6.4.1 R-Q Model and Linear MAD Prediction Model

In [81], Chiang and Zhang first introduced a new rate control scheme based on a quadratic formulation of the rate distortion function. This single pass technique has been widely used for MPEG-4 single and multiple video objects rate control [81, 82, 96]. For its simplicity and efficiency, it is also adopted in the JVT-G012 [77] rate control for H.264/AVC. The standard form of this quadratic R-Q model is given by

$$R_{texture,i} = S_i \times \left(\frac{X_{1,i}}{Q_i} + \frac{X_{2,i}}{Q_i^2} \right) \quad (6.10)$$

where $R_{texture,i}$ is the bit rate for the texture information of the current i th coding unit. (e.g., frame or MB). S_i and Q_i are its coding complexity measure and quantization step size, respectively. $X_{1,i}$ and $X_{2,i}$ are two model parameters which will be updated using linear regression method as follows [81]

$$X_2 = \frac{n \sum_{i=1}^n K_i - \left(\sum_{i=1}^n Q_i^{-1} \right) \left(\sum_{i=1}^n Q_i \cdot K_i \right)}{n \sum_{i=1}^n Q_i^{-2} - \left(\sum_{i=1}^n Q_i^{-1} \right)^2} \quad (6.11)$$

$$X_1 = \frac{\sum_{i=1}^n (Q_i \cdot K_i - X_2 \cdot Q_i^{-1})}{n} \quad (6.12)$$

where n is the number of coding units observed in the past sliding window, $K_i = R_i / S_i$ in which S_i is usually denoted by the MAD of this coding unit. Besides this quadratic R-Q model, the MAD has been used as the coding complexity measure in many other rate control schemes [85, 86, 87, 88]. Due to the fact that the actual MAD is not available before the QP calculation, a linear model for MAD prediction is proposed to solve this dilemma [77]. According to this model, the MAD of current coding unit can be predicted by the MAD of the coding unit at the co-located position in the previous frame. For example, let $PMAD_i$ and MAD_{i-1} be the predicted MAD for the current frame i and the actual MAD of the previous frame $i-1$, respectively. The linear prediction model is given as

$$PMAD_i = a_1 \times MAD_{i-1} + a_2 \quad (6.13)$$

where a_1 and a_2 are two parameters that will be updated after coding each frame. This linear prediction model is implicitly based on an assumption that the characteristics of video frames change gradually. In other words, it assumes that the increasing or decreasing tendency for MAD is likely to be the same within a short period of time. This assumption holds for those scenes with slow motions or nearly stationary scenes.

However, it fails when scene change occurs or there are lots of complex motion contents. In order to follow the non-stationary characteristics of real-world videos and to improve the frame MAD prediction accuracy, we propose an adaptive frame MAD prediction model in which the statistics of the current original frame is considered. Moreover, our predicted frame MAD will be further used as the frame complexity measure to adjust the frame level bit allocation.

6.4.2 Histogram Based Frame MAD Prediction Model

Histogram based distance measures have been widely used for temporal segmentation and frame rate determination in video coding [90, 100]. Lee *et al.* in [100] considered several different approaches for temporal segmentation including difference of histogram (DOH), histogram of difference image (HOD) and the motion compensation error (*i.e.*, MAD). The DOH of current frame n is defined by the mean absolute sum of the histogram difference between frame n and its reference frame $n-1$. It is calculated as follows

$$DOH_n = \frac{1}{N} \sum_{i=0}^{q-1} |h_n(i) - h_{n-1}(i)| \quad (6.14)$$

where h_n and h_{n-1} are luminance histograms for frames n and $n-1$, q is the number of levels, N is the total number of pixels in one frame. As a frame distance measure, the advantage of DOH is that it is insensitive to local motion activities and is better for detecting global changes. On the contrary, HOD is more sensitive to local motions than DOH. It is defined by

$$HOD_n = \frac{1}{N} \sum_{i \in [-T, T]} hod(i) \quad (6.15)$$

where hod is the histogram of the difference image between frame n and its reference frame $n-1$, T is a threshold value which regulates the sensitivity of the change between

Table 6.2. Correlation coefficients between actual frame MAD and the frame distance FD for different sequences

Suzie	Stefan	News	Carphone	Foreman	Trevor
0.930	0.974	0.890	0.948	0.883	0.931
0.910	0.929	0.845	0.938	0.849	0.892

two co-located pixels. The definition of N remains the same. To combine the advantages of both DOH and HOD, we propose to use a new frame distance measure FD_n which is a weighted combination of DOH and HOD as follows

$$FD_n = \omega \cdot DOH_n + (1 - \omega) \cdot HOD_n \quad (6.16)$$

where ω is a weighting factor with a typical value of 0.5 in our scheme. As pointed out in [100], although MAD is more ideal to be used as frame distance measure than the histogram based approaches, it is computationally too expensive to be realized.

Nevertheless, our experimental observations show that there exists a strong correlation between MAD and the histogram based distance measure. The correlation coefficients between the actual frame MAD and our frame distance measure FD for various sequences are tabulated in Table 6.2. When conducting this simulation, we set the values of q and T in (6.14) and (6.15) to 256 and 10 respectively. As can be seen, the correlations between these two measures are very strong which implies that the histogram based FD could be used to estimate the actual frame MAD. In our proposed scheme, the new frame MAD prediction model is given by

$$PMAD_i = MAD_{i-1} \times \left(\alpha + \frac{FD_i}{FD_{i-1}} \cdot (1 - \alpha) \right) \quad (6.17)$$

where $PMAD_i$ is the predicted MAD of current frame i , MAD_{i-1} is the actual MAD of the previous frame $i-1$, FD_i and FD_{i-1} are histogram based frame distances for the current and previous frames calculated using (6.16). Note that, we also introduce a

Figure 6.8. Abrupt scene change detection using FD

forgetting factor α with a constant value of 0.3 in our model. Compared to the linear MAD prediction model, our model not only depends on the previous MAD value but also takes the statistic of current frame into consideration. This can greatly improve the estimation accuracy especially when scene change occurs or there are complex motions. In addition, we propose to use the frame distance measure FD for abrupt scene change detection. In this work, if the following condition is satisfied

$$FD_i > TH \times FD_{i-1} \quad (6.18)$$

then we consider frame i as an abrupt scene change frame and its quantization parameter will be determined using our adaptive intra R-Q model. In this work, we are focusing on a simple and efficient scheme to detect abrupt scene changes for our rate control applications. Other existing scene change detection algorithms [120, 121] can also be used here. However, to avoid extra computations, we just propose to detect abrupt scene changes

Table 6.3. Comparison of MAD prediction accuracy

Sequence	Average frame MAD prediction error (%)		Prediction accuracy gain (%)
	Linear model	Proposed	
Foreman	3.18	2.78	12.6
Stefan	2.82	2.39	15.2
Trevor	6.46	3.76	41.8
Carphone	2.93	2.66	9.2
Suzie	3.74	2.82	24.6
News	4.93	3.70	24.9

based on the frame distance measure FD using Eq. (6.16). As an example, Fig. 6.8 illustrates the FD_i / FD_{i-1} for a synthesis video sequence which is generated by cascading the first 30 frames of five QCIF sequences including *Foreman*, *Trevor*, *Stefan*, *Carphone* and *Suzie*. Based on this observation and other experimental results, we found that $TH=3$ would be a suitable value to detect abrupt scene changes as can be seen from Fig. 6.8. Therefore the value of TH is empirically set to 3 in our rate control scheme.

6.4.3 MAD Prediction Model Performance Evaluation

In order to evaluate the performance of our proposed frame MAD prediction model, we have conducted two groups of experiments. In each group, seven standard QCIF sequences were coded based on the JM11 encoder [101] at the bit rate $u=32$ kbps and with the frame rate $Fr=15$ fps. In the first group, the original linear MAD prediction model as in (6.13) was adopted in the frame level rate control. Our proposed frame distance based model was applied in the second group. For both groups, the frame skipping control was disabled. We have recorded all of the predicted frame MAD values and the actual MAD values for both methods and the average frame MAD prediction errors are listed in Table 6.3. It can be seen from the table that the frame MAD prediction accuracy has been

Figure 6.9. MAD prediction error comparisons for *News*

consistently improved for all test sequences. Up to 41% estimation errors have been reduced for *Trevor* sequence and on average our proposed scheme has achieved 25% improvement over the existing linear prediction model. Fig. 6.9 and Fig. 6.10 further illustrate the frame by frame comparison for *News* and *Trevor*. Note that, there are two abrupt background changes for sequence *News* and one scene transition in *Trevor*. As expected, the traditional linear prediction model is unable to estimate frame MAD accurately when the scene contents are changing. On the other hand, since our model parameter FD_i is based on the statistic of current frame, the performance of our model will not be degraded seriously when scene change occurs. As a result, our histogram based MAD prediction model is more accurate and robust than the linear model. Note that, if we relax the one-pass constraint, we can use the actual MAD obtained with a fixed inter mode and better performance will be achieved at the cost of increased computational complexity.

Figure 6.10. MAD prediction error comparisons for *Trevor*

6.4.4 Improved Frame Level Bit Allocation Scheme

Generally, the frame level bit allocation in JVT-G012 [77] is based on a fluid flow traffic model and the linear tracking theory [80]. The target number of bits of current frame is determined by the frame rate, the current buffer level, the target buffer level and the available channel bandwidth. Firstly, the GOP level bit allocation is conducted as follows. According to [94], a typical GOP structure of IPPP... is used for low-delay applications. The total number of remaining bits $T_r(j)$ after encoding the $(j-1)$ th frame is given by

$$T_r(j) = \begin{cases} \frac{u}{F_r}(N_p + 1) & j = 1 \\ T_r(j-1) - b(j-1) & j = 2, 3, \dots, N_p + 1 \end{cases} \quad (6.19)$$

where u is the available channel bandwidth, F_r is the frame rate, N_p is the total number of P frames and $b(j-1)$ is the actual number of bits for encoding frame $(j-1)$. Note that, the buffer level (or buffer occupancy) is calculated after coding each frame by

$$B(j+1) = B(j) + b(j) - \frac{u}{F_r} \quad (6.20)$$

Secondly, the target number of bits for the current frame is determined based on both the buffer status and the number of remaining bits. As in [77], the target buffer level $Tbl(j)$ for each P frame is defined by

$$Tbl(j) = Tbl(j-1) - \frac{0.4 \times Bs}{N_p - 1} \quad j \geq 3 \quad (6.21)$$

where Bs is the buffer size. Let $T_{buffer}(j)$ be the target number of bits calculated according to the buffer status which is given by

$$T_{buffer}(j) = \frac{u}{F_r} - \gamma(B(j) - Tbl(j)) \quad (6.22)$$

where γ is a weighting factor with a constant value of 0.5. Besides, the average number of remaining bits for P frames is as follows

$$T_{ave}(j) = T_r(j) / N_r \quad (6.23)$$

where N_r represents the number of remaining P frames to be coded. Based on (6.22) and (6.23), the target number of bits $T(j)$ allocated to the current P frame in [77] is calculated by a weighted combination of two components: $T_{ave}(j)$ and $T_{buffer}(j)$

$$T(j) = \beta \cdot T_{ave}(j) + (1 - \beta) \cdot T_{buffer}(j) \quad (6.24)$$

where β is a predefined weighting factor in the range of (0.5, 1.0). As can be seen from (6.24), the remaining bits are allocated to the current frame without considering its frame complexity. This is not desirable for achieving a consistent quality video. Ribas-Corbera *et al.* in [89, 94] proposed an optimal frame level bit allocation scheme for H.263. In their

method, there exists a linear relationship between the optimal target bits and the frame complexity measure σ^2 which is the variance of the DCT coefficients of the motion compensated residue frame. Different from using the frame variance, Xie *et al.* [85, 86] adopted the actual frame MAD as a frame complexity measure and they introduced a scaling factor to adjust the target bits for current frame as follows

$$T_n = C \cdot \sqrt{MAD_n / \overline{MAD}_{n-1}} \quad (6.25)$$

where T_n is the target bit allocated for the current frame, C is the average output bit rate which is defined as $C = \text{bitrate}/\text{framerate}$, MAD_n is the actual MAD of the current frame and $\overline{MAD}_{n-1}$ denotes the average MAD of all previous encoded frames.

Due to the fact that the statistic of the current residue frame is not available in the frame bit allocation stage in H.264/AVC rate control, we propose to use the predicted frame MAD by our improved prediction model as the frame complexity measure. Similar to previous work, our objective is to allocate more bits to high complexity frames and fewer bits to low complexity frames in order to achieve a smoother video quality. In our proposed scheme, the $T_{ave}(j)$ in (6.24) is replaced by a weighted factor $\hat{T}_{ave}(j)$ which is given by

$$\hat{T}_{ave}(j) = T_{ave}(j) \times \frac{PMAD_j}{\overline{MAD}_{j-1}} \quad (6.26)$$

where $PMAD_j$ is the predicted MAD for the current frame i using our frame MAD prediction model (6.17) and $\overline{MAD}_{j-1}$ is the average MAD of all previous frames.

Finally, in order to comply with the hypothetical reference decoder (HRD) buffer requirement of H.264/AVC [76], the frame target bits will be bounded by

$$\begin{aligned} T(j) &= \max \{L(j), T(j)\} \\ T(j) &= \min \{U(j), T(j)\} \end{aligned} \quad (6.27)$$

Figure 6.11. Block diagram of the rate control framework for H.264/AVC

where $L(j)$ and $U(j)$ are lower bound and upper bound for HRD consideration respectively and they are calculated by

$$L(j) = \begin{cases} \frac{u}{F_r} & j = 1 \\ L(j-1) + \frac{u}{F_r} - b(j-1) & otherwise \end{cases} \quad (6.28)$$

$$U(j) = \begin{cases} (0.8 \times Bs) \cdot \varpi & j = 1 \\ U(j-1) + \left(\frac{u}{F_r} - b(j-1) \right) \cdot \varpi & otherwise \end{cases} \quad (6.29)$$

where ϖ is a constant with a typical value of 0.9.

Our proposed frame level rate control algorithm is summarized as follows:

- 1) Before encoding the current frame, compute the histogram based frame distance FD and the predicted frame MAD $PMAD$ using (6.16) and (6.17) respectively. Based on the buffer status, the available channel bandwidth and the predicted complexity of the current frame, the target frame bits T can be calculated by (6.24) and (6.26). Then it will be further bounded according to the HRD consideration.
- 2) Use (6.18) to check whether this frame is a scene change frame. If it is a scene change frame, then it will be coded as an intra-frame using our intra R-Q model in step 4. Otherwise go to step 3.
- 3) Based on the $PMAD$ and the target frame bits T , the QP for the current frame is derived from the quadratic R-Q model in (6.10). Go to step 5.
- 4) Calculate the frame gradient value G , determine the QP for the current frame using our adaptive intra R-Q model (6.5) and (6.6). Go to step 5.
- 5) Encode the current frame using the QP from step 3 or step 4.
- 6) Update the R-Q model parameters and update the buffer occupancy. Check if the buffer occupancy is above 80% of B_s . If it is true, skip the next frame (or frames) until the buffer occupancy is below 80%. Go to step 1 and proceed to the next frame.

Generally, our proposed frame level rate control method can be extended to the basic unit level (MB level or group of contiguous MBs as defined in [77]) rate control by incorporating the basic unit level bit allocation scheme similar to [77]. The block diagram of our rate control frame work is illustrated in Fig. 6.11. Compared to the traditional frame level rate control in JVT-G012 [77], only a limited number of operations are added for calculating the frame gradient and frame histograms. The overall computational complexity increment for the encoder is negligible. Therefore our proposed scheme is

Table 6.4. Comparisons of actual bitrate, frame skipping, average luminance PSNR and encoding time between JM (JVT-G012) rate control and our proposed rate control method.

Sequence (bitrate, framerate)	Actual bitrate (kbps)		No. of skipped frames		PSNR (dB)			Encoding time (sec.)	
	JM	Proposed	JM	Proposed	JM	Proposed	Gain	JM	Proposed
Foreman QCIF (24kbps, 10fps)	24.08	24.07	9	7	33.24	33.33	+ 0.11	68.3	68.4
Stefan QCIF (128kbps, 15fps)	127.89	127.75	0	0	32.32	32.32	0	82.0	82.4
Trevor QCIF (96kbps, 30fps)	95.90	95.95	9	4	35.47	35.77	+ 0.30	68.7	68.9
Suzie QCIF (32kbps, 10fps)	31.87	32.00	5	0	37.61	37.78	+ 0.17	66.5	66.9
Suzie QCIF(128kbps, 30fps)	128.53	128.29	1	1	37.89	38.08	+0.19	68.5	68.8
Foreman CIF (256kbps, 30fps)	255.84	256.12	0	0	35.08	35.09	+ 0.01	267.7	268.1
News CIF(400kbps, 30fps)	400.73	400.75	0	0	41.04	41.07	+0.03	265.3	266.0
Comb2 QCIF (24kbps, 10fps)	23.91	24.10	12	4	34.06	34.40	+ 0.34	67.5	67.9

highly desirable for real-time low-delay video coding. In the next section, the effectiveness of the proposed method will be verified by our extensive experiments.

6.5 Experimental Results

We have implemented our proposed algorithm to the H.264/AVC reference software JM11 [101]. The original rate control algorithm JVT-G012 [77] in JM11 was used as a benchmark for comparison. The experiments were conducted using the first 150 frames of four QCIF (176x144) test video sequences and two CIF (352x288) sequences with various target bit rates u , and frame rates Fr . We generated a new sequence *Comb2* for scene change by cascading the first 50 frames of *Foreman*, *Carphone* and *Silent*. In our experiments, the frame coding structure was IPPP with GOP size of 150. This format is very typical in low delay applications [93,94]. Similarly, our buffer size (B_s) was set to $1.25 \times u / Fr$ and the buffer overflow threshold was set to $0.8 \times B_s$ as same as [93,94]. The initial QP value of 38 was used for all test videos. The RDO was enabled in motion estimation and CABAC was adopted for entropy coding. In motion estimation, we set the

Table 6.5. Comparisons of PSNR variation and frame bits mismatch between JM (JVT-G012) rate control and our proposed rate control method.

Sequence (bitrate, framerate)	PSNR std. deviation			Frame bits mismatch ratio (%)		
	JM	Proposed	Reduction (%)	JM	Proposed	Reduction (%)
Foreman QCIF (24kbps, 10fps)	1.64	1.50	9.6	112	89	20.5
Stefan QCIF (128kbps, 15fps)	1.41	1.31	7.1	29	27	6.9
Trevor QCIF (96kbps, 30fps)	3.02	2.62	13.2	29	26	10.3
Suzie QCIF (32kbps, 10fps)	2.19	1.70	22.4	82	30	63.4
Suzie QCIF(128kbps, 30fps)	1.74	1.66	4.6	14	14	0
Foreman CIF (256kbps, 30fps)	0.91	0.82	9.9	20	18	10.0
News CIF(400kbps, 30fps)	1.90	1.80	5.3	11	10	9.1
Comb2 QCIF (24kbps, 10fps)	2.08	1.60	23.1	110	42	61.8

number of reference frames to be one and selected the search range of 16. We have configured all other encoder parameters to be the same for both methods. All the tests were conducted using a desktop PC with Pentium IV 2.4GHz CPU and 512 MB RAM.

Table 6.4 tabulates test results of the actual bit rate, number of skipped frames, the average luminance PSNR and the encoding time for both algorithms. Note that, for skipped frames, the PSNR was calculated by using the previous displayed frame for the current frame. From the table we find that both algorithms can achieve accurate overall output bit rate and the computational complexity increment by our scheme is negligible. It is observed that by using our proposed frame level rate control method, the overall PSNR have been improved for most sequences by up to 0.34 dB. As can be seen, the number of skipped frames has been significantly reduced by adopting our scheme. This is due to our enhanced frame MAD prediction model and frame level bit allocation scheme. In addition, our adaptive intra R-Q model with abrupt scene change detection method also reduces the chance of buffer overflow during scene change frames. Table 6.5 shows the PSNR variation and frame bits mismatch for both rate control schemes. The proposed algorithm achieves on average 12% reduction in PSNR deviations for all sequences. This

(a)

(b)

(c)

Figure 6.12. Frame by frame comparisons for *Trevor* sequence (96kbps, 30fps, QCIF):
(a) Luminance PSNR (b) Buffer fullness (c) Frame bits mismatch

indicates that our rate control scheme can provide a smoother visual quality for the encoded video with less quality fluctuations and motion jerkiness. Furthermore, compared to the original method, the frame level bits mismatch has been decreased by 23% on average for all test videos. In other words, our scheme achieves more accurate frame bit rate control than the traditional algorithm. As an example, Fig. 6.12 illustrates frame by frame comparisons for *Trevor* sequence. It can be seen that, our algorithm achieves improved performance over the traditional method by providing reduced PSNR fluctuations, more stable buffer fullness and reduced frame level rate control errors.

6.6 Conclusion

In this chapter, we have presented an efficient frame complexity based rate control scheme for H.264/AVC video coding. First, we investigated the intra-frame R-Q model for H.264/AVC coding. We proposed to use frame gradient as the complexity measure in R-Q modeling for intra-coded frames. This R-Q model can be updated adaptively during the encoding process. Our simulation results show that our model is very robust and efficient for intra-frames regardless of scene content changes. Second, an improved frame MAD prediction model and a new frame bit allocation scheme have been discussed. The difference between the proposed frame MAD prediction model and the traditional linear prediction model is that the statistic of current frame (*i.e.*, histogram information) has been considered. Based on this enhanced MAD prediction model, we further proposed a frame bit allocation scheme using predicted MAD as the frame complexity measure. Our objective is to allocate more bits to high complexity frames and fewer bits to low complexity frames in order to achieve smoother visual quality. Extensive experimental results show that the proposed scheme can enhance the overall performance of the

existing rate control for H.264/AVC. Compared to the traditional method for JVT, our proposed scheme achieves improved average PSNR with smaller PSNR variation, more accurate frame level bit rate control and reduced buffer overflows. Thus, it is very suitable for real-time H.264/AVC applications.

Chapter 7

Conclusions and Extensions

7.1 Summary of the Research

In this research, our focus has been on efficient motion compensation and rate control methods to achieve better tradeoff between video coding complexity and quality as well as better utilization of the available channel bandwidth. We have reviewed the basics of motion compensation and rate control techniques for a generic video compression system in Chapter 2. Both technologies are indispensable for the efficient video communication applications. Our first research topic was on the motion compensation techniques including motion estimation and mode decision. The motion compensation is very essential in reducing the temporal redundancies among source video frames. In addition, it is the most computation intensive part in the video coding system.

To cope with the high computational complexity problem for motion compensated video coding, firstly we have proposed two fast algorithms for block matching motion estimation in Chapter 3. Both methods are based on the observations of statistical

characteristics of motion vector distribution. The first approach, *i.e.*, E3SS, belongs to the fast search category in which only a part of the candidates in the search window are examined. It utilizes effective search pattern and search strategy to enhance the block matching accuracy with low computational complexity. The second approach belongs to the fast matching category. It is a novel partial distortion search algorithm which uses a predictive search area for motion estimation. Simulation results show that the proposed algorithm achieves significant computation reduction compared with other PDS based algorithms with negligible degradation in picture quality. Besides these two fast motion estimation algorithms, we proposed a new block matching criterion for motion estimation in Chapter 4. This new criterion, *i.e.*, SCMAD, aims at eliminating the mismatch between motion estimation and the following transform coding. The main advantage of SCMAD over MAD is that it can further distinguish candidate blocks by taking the smoothness constraint into consideration thus leading to a better decision in terms of smaller bit rate.

In Chapter 5 we proposed an efficient inter mode decision algorithm for H.264/AVC standard. By using seven different block types in motion compensation, the R-D performance has been greatly improved compared with the FSBM scheme. On the other hand, the computational complexity of H.264/AVC encoder has been dramatically increased due to the various block types used. Based on efficient MB classification scheme, our inter mode decision method achieves significant computation savings of the RDO based mode decision for homogeneous MBs by avoiding the use of small sized block types.

Our second research topic was on the frame level rate control for real-time H.264/AVC video coding. We proposed two frame complexity based schemes to enhance the existing H.264/AVC rate control method in Chapter 6. First, we investigated the intra-frame R-Q model for H.264/AVC coding. We developed an adaptive intra-frame R-Q model using frame gradient as the complexity measure. This R-Q model is very robust

and efficient for intra-frames regardless of scene content changes. Second, we proposed an improved frame MAD prediction model to enhance the traditional linear prediction model which is used to solve the typical *chicken and egg* dilemma in H.264/AVC RDO process. By taking the frame histogram information into consideration, this new MAD prediction model is more adaptive to scene changes. Based on this enhanced MAD prediction model, we further proposed a new frame level bit allocation scheme using predicted frame MAD as the frame complexity measure. In general, more bits will be allocated to the frames with higher complexity and vice versa. Experimental results show that the overall performance of the traditional rate control method for H.264/AVC can be improved in terms of target bits mismatch and the visual quality of reconstructed video.

7.2 Future Extensions

In addition to the motion compensation and rate control approaches discussed in this thesis, there are some interesting extensions for this work which are presented in the following.

- **Fast intra mode decision for H.264/AVC video coding**

Besides the seven block types used in the inter mode decision for H.264/AVC, as mentioned in Chapter 5, this new standard also provides a number of prediction patterns for intra mode decision [76]. Specifically, nine prediction modes for Intra 4×4 blocks and four prediction modes for Intra 16×16 blocks are enabled in the standard. The original implementation of this RDO based mode decision for intra mode is based on exhaustive search for all prediction patterns and finally the best mode which can produce the minimum RDO cost will be chosen. In order to further reduce the high computational complexity of this exhaustive search for intra mode decision, the basic

concept of our fast inter mode decision algorithm can potentially be used to achieve this goal. By conducting proper classification of all MBs in each frame, we can save the computation by discarding small sized prediction patterns for those MBs which contain relatively simple texture.

- **Efficient mode decision for scalable video coding (SVC)**

The proposal of scalable video coding (SVC) [110] has been adopted as an extension of the H.264/AVC standard recently. In the spatial and SNR scalability, according to the layered coding structure, the prediction of each enhancement layer is based on the base layer and previous enhancement layers. Same as the H.264/AVC, the SVC determines the best coding mode for each MB from the base layer or enhancement layers by examining every possible mode. Therefore the computational complexity of SVC encoder is extremely high and it is worthwhile to design an efficient mode decision scheme for SVC video coding. For the base layer coding, our inter mode decision scheme can be used directly to SVC. For enhancement layers, the coding mode for each MB can possibly be predicted from the coding modes of its collocated MBs in its reference coding layers.

- **Subjective quality guided bit allocation for MB level rate control**

Currently, our proposed rate control algorithm in Chapter 6 is frame level based in which all MBs in a frame are coded using the same QP. It is possible to integrate our frame level bit allocation with the MB level bit allocation scheme. By considering the perceptual distortion sensitivity [109] as the coding complexity measure for MBs within a frame, it is reasonable to allocate more bits to those MBs which are more sensitive to distortions. On the contrary, fewer bits may be allocated to those MBs which can tolerate more perceptual distortions. Moreover, based on the MB classifica-

tion and grouping, we can update the R-Q models for different groups of MBs respectively. We believe that by integrating this adaptive MB level rate control approach, the subjective quality of reconstructed video can be further improved.

Besides the above mentioned extensions, it is also a promising direction to consider the overall optimization problem for a specified codec such as an H.264 encoder. This global optimization aims at achieving better tradeoffs between multiple factors: complexity, visual quality and bit rate. Such power-rate-distortion oriented optimization problem is extremely useful for real-time video applications [112].

Publications

Journal Publications

1. Xuan Jing, Ce Zhu and Lap-Pui Chau, "Smooth constrained motion estimation for video coding," *EURASIP Signal Processing*, vol. 83, no. 3, pp. 677-680, Mar. 2003.
2. Xuan Jing and Lap-Pui Chau, "An efficient three-step search algorithm for block motion estimation," *IEEE Transactions on Multimedia*, vol 6, no. 3, pp. 435-438, Jun. 2004.
3. Xuan Jing and Lap-Pui Chau, "Fast approach for H.264 inter mode decision", *Electronics Letters*, vol. 40, no. 17, pp. 1050-1052, Aug. 2004.
4. Xuan Jing and Lap-Pui Chau, "Partial distortion search algorithm using predictive search area for fast full-search motion estimation," *IEEE Signal Processing Letters*, vol. 14, no. 1, pp. 840-843, Nov. 2007.
5. Xuan Jing, Lap-Pui Chau and Wan-Chi Siu, "Frame complexity based rate-quantization model for H.264/AVC intraframe rate control," *IEEE Signal Processing Letters*, vol. 15, no. 1, pp. 373-376, 2008.

Conference Publications

1. Lap-Pui Chau and Xuan Jing, "Efficient three-step search algorithm for block motion estimation in video coding," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'03*, Hong Kong, vol. 3, pp. 421-424, Apr. 2003.
2. Xuan Jing, Ce Zhu and Lap-Pui Chau, "Smooth constrained block matching criterion for motion estimation," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'03*, Hong Kong, vol. 3, pp. 661-664, Apr. 2003.
3. Xuan Jing and Lap-Pui Chau, "An efficient inter mode decision approach for H.264 video coding", in *Proceedings of IEEE International Conference on Multimedia and Expo, ICME'04*, Taipei, vol. 2, pp. 1111-1114, Jun. 2004.
4. Xuan Jing and Lap-Pui Chau, "A novel intra-rate estimation method for H.264 rate control," in *Proceedings of IEEE International Symposium on Circuits and Systems, ISCAS'06*, Greece, pp. 5019-5022, May 2006.
5. Xuan Jing and Lap-Pui Chau, "Improved frame level MAD prediction and bit allocation scheme for H.264/AVC rate control," in *Proceedings of IEEE International Symposium on Circuits and Systems, ISCAS'07*, USA, pp. 3639-3642, May 2007.

Bibliography

- [1] ITU-T SG15, "Video codec for audiovisual services at $p \times 64$ kbit/s," *ITU-T recommendation H.261 Version 3*, Mar. 1993.
- [2] ITU-T SG15, "Video coding for low bitrate communication," *ITU-T recommendation H.263*, May 1996.
- [3] JTC 1/SC 29, "Information technology - coding of moving pictures and associated audio for digital storage media at up to 1.5 Mbit/s – part 2: Video," *ISO/IEC 11172-2(MPEG-1 Video)*, 1993.
- [4] JTC 1/SC 29, "Information technology – generic coding of moving pictures and associated audio information: Video," *ISO/IEC 13818-2/ITU-T H.262*, 1996.
- [5] ISO/IEC 14496-2, Coding of Audio-Visual Objects – Part 2: Visual, 2001.
- [6] U. Reimers, "Digital video broadcasting," *IEEE Communications Magazine.*, vol. 36, no. 6, pp. 104-110, Jun. 1998.
- [7] Advanced Television Systems Committee, "ATSC digital television standard," Sep. 1995.
- [8] A.J. Stienstra, "Technologies for DVB services on the Internet," in *Proceedings IEEE*, vol. 94, no. 1, pp. 228-236, Jan. 2006.
- [9] B. Zeng, R. Li and M.L. Liou, "Optimization of fast block motion estimation algorithms," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 6, pp. 833-844, Dec. 1997.
- [10] F. Dufaux, F. Moscheni, "Motion estimation techniques for digital TV: A review and a new contribution", in *Proceedings IEEE*, vol. 83, no. 6, pp. 858-876, Jun. 1995.

- [11] J.R. Jain, A.K. Jain, "Displacement measurement and its application in interframe image coding," *IEEE Trans. Commun.*, vol. 29, pp. 1799-1806, Dec. 1981.
- [12] T. Koga, K. Iinuma, A. Hirano, Y. Iijima and T. Ishiguro, "Motion compensated interframe coding for video conferencing," in *Proc. Nat. Telecommun. Conf.*, New Orleans, LA, Nov 29-Dec. 3, 1981, pp. G5.3.1-5.3.5
- [13] A. Puri, H.M. Hang, and D.L. Schilling, "An efficient block matching algorithm for motion-compensated coding," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, pp. 1063-1066, 1987.
- [14] M. Ghanbari, "The cross-search algorithm for motion estimation," *IEEE Trans. Commun.*, vol. COM-38, pp. 950-953, Jul. 1990.
- [15] R. Li, B. Zeng, and M.L. Liou, "A new three-step search algorithm for block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, no. 4, pp. 438-442, Aug. 1994.
- [16] L.M. Po and W.C. Ma, "A novel four-step search algorithm for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 3, pp. 313-317, Jun. 1996.
- [17] L.K. Liu and E. Feig, "A block-based gradient descent search algorithm for block motion estimation in video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 4, pp. 419-422, Aug. 1996.
- [18] S. Zhu and K.K. Ma, "A new diamond search algorithm for fast block-matching motion estimation," *IEEE Trans. Image Processing.*, vol. 9, no. 2, pp. 287-290, Feb. 2000.
- [19] J.Y. Tham, S. Ranganath, M. Ranganath and A.A. Kassim, "A novel unrestricted center biased diamond search algorithm for block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 4, pp. 369-377, Aug. 1998.
- [20] C. Zhu, X. Lin and L.P. Chau, "Hexagon-based search pattern for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 5, pp. 349-355, May 2002.
- [21] Y.L. Chan and W.C. Siu, "An efficient search strategy for block motion estimation using image features," *IEEE Trans. Image Processing.*, vol. 10, no. 8, pp. 1223-1238, Aug. 2001.

- [22] D.K. Lim and Y.S. Ho, "A fast block matching motion estimation algorithm based on statistical properties of object displacement," in *TENCON'98*, Vol. 1, pp. 138-141, 1998.
- [23] C.K. Cheung and L.M. Po, "Normalized partial distortion search algorithm for block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 3, pp. 417-422, Apr. 2000.
- [24] J.B. Xu, L.M. Po and C.K. Cheung, "Adaptive motion tracking block matching motion algorithms for video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 7, pp. 1025-1029, Oct. 1999.
- [25] M. Song, A. Cai and J. Sun, "Motion estimation in DCT domain," in *ICCT'96, IEEE Int. Conf. on Communication Technology*, vol. 2, pp. 670-674, 1996.
- [26] S. Kappagantula and K.R. Rao, "Motion compensated interframe image prediction," *IEEE Trans. Communication.*, vol. COM-33, no. 9, pp. 1011-1015, Sep. 1985.
- [27] H. Gharavi and M. Mills, "Block-matching motion estimation algorithm-New results," *IEEE Trans. Circuits and Systems.*, vol. 37, no. 5, pp. 649-651, May 1990.
- [28] M.J. Chen, L.G. Chen, T.D. Chiueh and Y.P. Lee, "A new block-matching criterion for motion estimation and its implementation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, no. 3, Jun. 1995.
- [29] F. Moschetti, "A statistical approach to motion estimation," *Ph.D Thesis, EPFL*, 2001.
- [30] D.S. Turaga and T. Chen, "Estimation and mode decision for spatially correlated motion sequences," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 10, pp. 1098-1107, Oct. 2001.
- [31] A. Ortega and K. Ramchandran, "Rate-distortion methods for image and video compression," *IEEE Signal Processing Magazine.*, vol. 15, no. 6, pp. 23-50, Nov. 1998.
- [32] 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.
- [33] V. Bhaskaran and K. Konstantinides, *Image and video compression standards – algorithms and architectures*. Kluwer Academic Publishers, 1995.
- [34] C.H. Hsieh, P.C. Lu, J.S. Shyn and E.H. Lu, "Motion estimation algorithm using interblock correlation," *Electron. Lett.*, vol. 26, no. 5, pp. 276-277, Mar. 1990.

- [35] C.D. Bei and R.M. Gray, "An improvement of the minimum distortion encoding algorithm for vector quantization," *IEEE Trans. Commun.*, vol. COM-33, pp. 1132-1133, Oct. 1985.
- [36] J.N. Kim and T.S. Choi, "A fast full-search motion-estimation algorithm using representative pixels and adaptive matching scan," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 7, pp. 1040-1048, Oct. 2000.
- [37] B. Tao and M.T. Orchard, "Gradient-based residual variance modeling and its applications to motion-compensated video coding," *IEEE Trans. Image Processing.*, vol. 10, no. 1, pp. 24-35, Jan. 2001.
- [38] C.H. Cheung and L.M. Po, "Adjustable partial distortion search algorithm for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 1, pp. 100-110, Jan. 2003.
- [39] I. Ismaeil, A. Docef, F. Kossentini, and R. Ward, "Efficient motion estimation using spatial and temporal motion vector prediction," in *Proc. IEEE Int. Conf. Image Processing*, 1999, pp. 70-74.
- [40] W.G. Hong and T.M. Oh, "Sorting-based partial distortion search algorithm for motion estimation," *Electron. Lett.*, vol. 40, no. 2, pp. 113-114, Jan. 2004.
- [41] B. Tao and M.T. Orchard, "Feature-accelerated block matching," in *Proc. SPIE*, vol. 3309, pp. 469-476, 1997.
- [42] J.C. Tsai, C.H. Hsieh, S.K. Weng and M.F. Lai, "Block-matching motion estimation using correlation search algorithm," *Signal Process.: Image Commun.*, vol. 13, pp. 119-133. 1998.
- [43] B. Liu and A. Zaccarin, "New fast algorithms for the estimation of block motion vectors," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, pp. 148-157, Apr. 1993.
- [44] Y.L. Chan and W.C. Siu, "New adaptive pixel decimation for block motion vector estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 1, pp. 113-118, Feb. 1996.
- [45] K. Lengwehasatit and A. Ortega, "Probabilistic partial-distance fast matching algorithms for motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 2, pp. 139-152, Feb. 2001.
- [46] K.L. Chung and L.C. Chang, "A new predictive search area approach for fast block motion estimation," *IEEE Trans. Image Processing.*, vol. 12, no. 6, pp. 648-652, Jun. 2003.

- [47] Y.K. Tu, J.F. Yang, Y.N. Shen and M.T.Sun, "Fast variable size block motion estimation using merging procedure with an adaptive threshold," in *Proc. Int. Conf. Multimedia and Expo*, vol.2, pp.789-792, Jul. 2003.
- [48] A. Chang, O.C. Au and Y.M. Yeung, "A novel approach to fast multi-block motion estimation for H.264 video coding," in *Proc. Int. Conf. Multimedia and Expo*, vol.1, pp.105-108, Jul. 2003.
- [49] P.Yin, H.C. Tourapis, A.M. Touapis and J.Boyce, "Fast mode decision and motion estimation for JVT/H.264," in *Proc. Int. Conf. Image Processing*, vol. 2, pp. 853-856, Sep. 2003.
- [50] K.P. Lim, S. Wu, D.J. Wu, S. Rahardja, X. Lin, F. Pan and Z.G. Li, "Fast inter mode selection", JVT-I020, 9th JVT meeting, San Diego, USA, Sep. 2003.
- [51] S.Y. Huang, J.R. Chen, J.S. Wang, K.R. Hsieh and H.Y. Hsieh, "Classified variable block size motion estimation algorithm for image sequence coding," in *Proc. Int. Conf. Image Processing*, vol.3, pp.736-740, Nov. 1994.
- [52] T. Wiegand and B. Girod, "Lagrange multiplier selection in hybrid video coder control," in *Proc. Int. Conf. Image Processing*, vol. 3, pp. 542-545, Oct. 2001.
- [53] T. Wiegand, M. Lightstone, D. Mukherjee, T.G. Campbell and S.K. Mitra, "Rate-distortion optimized mode selection for very low bit rate video coding and the emerging H.263 standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 2, pp. 182-190, Apr. 1998.
- [54] T. Wiegand, G.J. Sullivan, G. Bjontegaard and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560-576, Jul. 2003.
- [55] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini and G.J. Sullivan, "Rate-constrained coder control and comparison of video coding standards," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 688-702, Jul. 2003.
- [56] H.S. Malvar, A. Hallapuro, M. Karczewicz and L. Kerofsky, "Low-complexity transform and quantization in H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 598-603, Jul. 2003.
- [57] I. Rhee, G.R. Martin, S. Muthukrishnan and R.A. Packwood, "Quadtree-structured variable-size block-matching motion estimation with minimal error," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 1, pp. 42-50, Feb. 2000.

- [58] J. Zhang, M.O. Ahmad and M.N.S. Swamy, "Quadtree structured region-wise motion compensation for video compression," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 5, pp. 808-822, Aug. 1999.
- [59] J.W. Kim and S.U. Lee, "Video coding with R-D constrained hierarchical variable block size (VBS) motion estimation," *Journal of Visual Communication and Image Representation.*, vol. 9, no. 3, pp. 243-254, Sep. 1998.
- [60] C. Kim and C.-C.J. Kuo, "Fast intra/inter mode decision for H.264 encoding using a risk-minimization criterion," in *Proc. of the SPIE*, vol. 5558, pp. 536-546, 2004.
- [61] A. Ahmad, N. Khan, S. Masud and M.A. Maud, "Efficient block size selection in H.264 video coding standard," *Electron. Lett.*, vol. 40, no. 1, pp. 19-21, Jan. 2004.
- [62] M.H. Chan, Y.B. Yu and A.G. Constantinides, "Variable size block matching motion compensation with application to video coding," *IEE Proc. Communications, Speech and Vision*, vol. 137, no. 4, pp. 205-212, Aug. 1990.
- [63] D.T. Hoang, P.M. Long and J.S. Vitter, "Efficient cost measures for motion estimation at low bit rates," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 4, pp. 488-500, Aug. 1998.
- [64] C.K. Cheung, "Fast motion estimation techniques for video compression," Master's Thesis, City University of Hong Kong, Jul. 1998.
- [65] B.G. Haskell and J.O. Limb, "Predictive video encoding using measured subjective velocity," US Patent no. 3,632,865, Jan. 1972.
- [66] F. Rocca, "Television bandwidth compression utilizing frame-to-frame correlation and movement compensation," *Symp. Picture Bandwidth Compression*, New York, 1969.
- [67] B.K.P. Horn and B.G. Schunck, "Determining optical flow," *Artif. Intell.*, vol. 17, pp. 185-203, 1981.
- [68] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. Image Understanding Workshop*, pp. 121-130, 1981.
- [69] A. Netravali and J.D. Robbins, "Motion compensated television coding part I," *Bell Syst. Tech. J.*, vol. 58, no. 3, pp. 629-668, Apr. 1979.
- [70] D.R. Walker and K.R. Rao, "Improved pel-recursive motion compensation," *IEEE Trans. Commun.*, vol. COM-32, pp. 1128-1134, Oct. 1984.

- [71] J.S. Kim and R.H. Park, "A fast feature-based block matching algorithm using integral projections," *IEEE Journal on Selected Areas in Communications*, vol. SAC-10, no. 5, pp. 968-971, Jun. 1992.
- [72] H.261 Software codec: PVRG-P64, Portable Video Research Group.
- [73] ISO-IEC/JTC1/SC29/WG11/N0400, *Test Model 5*, Apr. 1993.
- [74] ITU-T/SG15, *Video Codec Test Model*, TMN8, Portland, Jun. 1997.
- [75] MPEG-4 Video Verification Model V18.0, Coding of Moving Pictures and Audio N3908, ISO/IEC, JTC1/SC29/WG11, Jan. 2001.
- [76] T. Wiegand and G.J. Sullivan, "Draft ITU-T recommendation H.264 and final draft international standard of joint video specification (ITU-T recommendation h.264|ISO/IEC 14496-10 AVC)," Joint Video Team of ISO/IEC JTC1/SC29/WG11 and ITU-T SG16/Q.6 Doc. JVT-G050 Pattaya, Thailand, Mar. 2003.
- [77] Z.G. Li, F. Pan, K.P. Lim, G.N. Feng, X. Lin and S. Rahardaj, "Adaptive basic unit layer rate control for JVT," Joint Video Team of ISO/IEC JTC1/SC29/WG11 and ITU-T SG16/Q.6 Doc. JVT-G012, Pattaya, Thailand, Mar. 2003.
- [78] G.J. Sullivan, T. Wiegand and K.P. Lim, "Joint model reference encoding methods and decoding concealment methods, Section 2.6: Rate Control," (JVT-I049), San Diego, Sep. 2003.
- [79] S. Ma, W. Gao and Y. Lu, "Rate control on JVT standard," Joint Video Team of ISO/IEC JTC1/SC29/WG11 and ITU-T SG16/Q.6 Doc. JVT-D030 Klagenfurt, Austria, Jul. 2002.
- [80] Z.G. Li, C. Zhu, N. Ling, X.K. Yang, G.N. Feng, S. Wu and F. Pan, "A unified architecture for real-time video coding systems," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 6, pp. 472-487, Jun. 2003
- [81] T. Chiang and Y.Q. Zhang, "A new rate control scheme using quadratic rate distortion model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 2, pp. 287-311, Apr. 1997.
- [82] H.J. Lee, T.H. Chiang and Y.Q. Zhang, "Scalable rate control for MPEG-4 video," *IEEE Trans. Circuits Syst. Video Technol.*, vol.10, no. 6, pp. 878-894, Sep. 2000.
- [83] Z. He, Y. Kim and S.K. Mitra, "Low-delay rate control for DCT video coding via ρ -domain source modeling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 8, pp. 928-940, Aug. 2001.

- [84] Z. He and T. Chen, "Linear rate control for JVT video coding," in *Proc. 2003 Int. Conf. Information Technology: Research and Education*, Newark, NJ, pp.11-13, Aug. 2003.
- [85] B. Xie and W. Zeng, "Sequence-based rate control for constant quality video," in *Proc. 2002 Int. Conf. Image Processing*, Rochester, NY, pp. 77-80, Sep. 2002.
- [86] B. Xie and W. Zeng, "A sequence-based rate control framework for consistent quality real-time video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 1, pp. 56-71, Jan. 2006.
- [87] S. Ma, W. Gao and Y. Lu, "Rate-distortion analysis for H.264/AVC video coding and its application to rate control," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 12, pp. 1533-1544, Dec. 2005.
- [88] Z. Zhang, G. Liu, H. Li and Y. Li, "A novel PDE-based rate-distortion model for rate control," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 11, pp. 1354-1364, Nov. 2005.
- [89] J. Ribas-Corbera and S. Lei, "A frame-layer bit allocation for H.263+," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 7, pp 1154-1158, Oct. 2000.
- [90] H. Song and C.-C.J. Kuo, "Rate control for low-bit-rate video via variable-encoding frame rates," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 4, pp. 512-521, Apr. 2001.
- [91] M. Jiang, X. Yi and N. Ling, "Improved frame-layer rate control for H.264 using MAD ratio," in *Proc. IEEE Int. Symp. Circuits Syst.*, Vancouver, BC, Canada, vol. III, pp. 813-816, May 2004.
- [92] J. Xu and Y. He, "A novel rate control for H.264," in *Proc. IEEE Int. Symp. Circuits Syst.*, Vancouver, BC, Canada, vol. III, pp. 809-812, May 2004.
- [93] M. Jiang and N. Ling, "Low-delay rate control for real-time H.264/AVC video coding," *IEEE Trans. Multimedia.*, vol. 8, no. 3, pp. 467-477, Jun. 2006.
- [94] J. Ribas-Corbera and S. Lei, "Rate control in DCT video coding for low-delay communications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 1, pp 172-185, Feb. 1999.
- [95] W. Ding and B. Liu, "Rate control of MPEG video coding and recording by rate-quantization modeling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 1, pp. 12-20, Feb. 1996.

- [96] K.N. Ngan, T. Meier and Z. Chen, "Improved single-video-object rate control for MPEG-4," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 5, pp. 385-393, May 2003.
- [97] F. Pan, Z.G. Li, K.P. Lim, X. Lin, S. Rahardja, D. Wu and S. Wu, "Adaptive intra-frame quantization for very low bit rate video coding," in *Proc. IEEE Int. Symp. Circuits Syst.*, Vancouver, BC, Canada, vol. III, pp 781-784, May 2004.
- [98] N. Kamaci, Y. Altunbasak and R.M. Mersereau, "Frame bit allocation for the H.264/AVC video coder via Cauchy-Density-Based rate and distortion models," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 8, pp 994-1006, Aug. 2005.
- [99] W. J. Kim, J. W. Yi and S. D. Kim, "A bit allocation method based on picture activity for still image coding," *IEEE Trans. Image Processing.*, vol. 8, no. 7, pp. 974-977, Jul. 1999.
- [100] J. Lee and B.W. Dickinson, "Temporally adaptive motion interpolation exploiting temporal masking in visual perception," *IEEE Trans. Image Processing.*, vol. 3, pp. 513-526, Sep. 1994.
- [101] [Online] Available: JM reference software
<http://iphome.hhi.de/suehring/tml/download>
- [102] X. Yang, W. Lin, Z. Lu, X. Lin, S. Rahardja, E. Ong and S. Yao, "Rate control for videophone using local perceptual cues," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 4, pp. 496-506, Apr. 2005.
- [103] Y. Sun, I. Ahmad, D. Li and Y.Q. Zhang, "Region-based rate control and bit allocation for wireless video transmission," *IEEE Trans. Multimedia.*, vol. 8, no. 1, pp. 1-10, Feb. 2006.
- [104] Y. Sun and I. Ahmad, "A robust and adaptive rate control algorithm for object-based video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 10, pp. 1167-1182, Oct. 2004.
- [105] L.J. Lin and A. Ortega, "Bit-rate control using piecewise approximated rate-distortion characteristics," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 4, pp. 446-459, Aug. 1998.
- [106] H.M. Hang and J.J. Chen, "Source model for transform video coder and its application—Part I: fundamental theory," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 2, pp. 287-297, Apr. 1997.

- [107]J.J. Chen and H.M. Hang, "Source model for transform video coder and its application—Part II: variable frame rate coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 2, pp. 299-311, Apr. 1997.
- [108]J.C. Tsai and C.H. Hsieh, "Modified TMN8 rate control for low-delay video communications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 6, pp. 864-868, Jun. 2004.
- [109]C.W. Tang, C.H. Chen, Y.H. Yu and C.J. Tsai, "Visual sensitivity guided bit allocation for video coding," *IEEE Trans. Multimedia.*, vol. 8, no. 1, pp. 11-18, Feb. 2006.
- [110]J. Reichel, H. Schwarz and M. Wien, Scalable Video Coding-Working Draft 4. Joint Video Team, ISO/IEC/ JTC1/SC29/WG11/JVT-Q201, Nice, France, Oct. 2005.
- [111]ITU-T/SG15, Video codec test model, TMN7, Nice, Feb. 1997.
- [112]L. Su, Y. Lu, F. Wu, S. Li and W. Gao, "Real-time video coding under power constraint based on H.264 codec," in *Proc. SPIE Visual Communications and Image Processing VCIP 2007*, vol. 6508, pp. 1-12, 2007
- [113]Z. Zhou, J. Xin and M.-T. Sun, "Fast motion estimation and inter-mode decision for H.264/MPEG-4 AVC encoding," *Journal of Visual Communications and Image Representation*, vol. 17, no. 2, pp. 243-263, Apr. 2006
- [114]T.-C. Chen, S.-Y. Chien, Y.-W. Huang, C.-H. Tsai, C.-Y. Chen, T.-W. Chen and L.-G. Chen, "Analysis and architecture design of an HDTV720p 30 frames/s H.264/AVC encoder," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 6, pp. 673-688, Jun. 2006
- [115]S. Saponara, K. Denolf, G. Lafruit, C. Blanch and J. Bormans, "Performance and complexity co-evaluation of the advanced video coding standard for cost-effective multimedia communications," *EURASIP Journal on Applied Signal Processing*, vol. 2, pp. 220-235, 2004
- [116]H. Wang, S. Kwong and C.W. Kok, "An efficient mode decision algorithm for H.264/AVC encoding optimization," *IEEE Trans. Multimedia*, vol. 9, no. 4, pp. 882-888, Jun. 2007
- [117]L. Yang, K. Yu, J. Li and S. Li, "An effective variable block-size early termination algorithm for H.264 video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 6, pp. 784-788, Jun. 2006

- [118]C. Grecos and M. Yang, "Fast mode prediction for the baseline and main profiles in the H.264 video coding standard," *IEEE Trans. Multimedia*, vol. 8, no. 6, pp. 1125-1134, Dec. 2006
- [119]J.H. Kim and B.G. Kim, "Fast block mode decision algorithm in H.264/AVC video coding," *Journal of Visual Communication and Image Representation*, vol. 19, no. 3, pp. 175-183, Oct. 2007
- [120]M.R. Naphade, R. Mehrotra, A.M. Ferman, J. Warnick, T.S. Huang and A.M. Tekalp, "A high-performance shot boundary detection algorithm using multiple cues," in *Proc. IEEE Int. Conf. Image Process.*, ICIP98, vol. 1, pp. 884-887, Oct. 1998
- [121]C.L. Huang and B.Y. Liao, "A robust scene-change detection method for video segmentation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 1281-1288, Dec. 2001