

**PERFORMANCE AND COMPUTATIONAL COMPLEXITY
OPTIMIZATION TECHNIQUES IN CONFIGURABLE VIDEO
CODING SYSTEM**

by

NYEONGKYU KWON

B.S., Han-Kuk Aviation University, Korea, 1988

M.S., Korea Advanced Institute of Science and Technology, Korea, 1990

A Dissertation Submitted in Partial Fulfillment of
the Requirements for the Degree of

DOCTOR OF PHILOSOPHY

In the Department of Electrical and Computer Engineering

We accept this thesis as conforming
to the required standard

© NYEONGKYU KWON, 2005

University of Victoria

*All rights reserved. This dissertation may not be reproduced in whole or in part, by
photocopying or other means, without the permission of the author.*

Supervisors: Dr. Peter F. Driessen and Dr. Pan Agathoklis

ABSTRACT

In order to achieve high performance in terms of compression ratio, most standard video coders have a high computational complexity. Motion estimation in sub-pixel accuracy and in model-based rate distortion optimization is approached from a practical implementation perspective; then, a configurable coding scheme is proposed and analyzed with respect to computational complexity and distortion. The proposed coding scheme consists of three coding modules: motion estimation, sub-pixel accuracy, and DCT pruning, and their control variables can take several values, leading to a significantly different coding performance.

The major coding modules are analyzed in terms of computational complexity and distortion (C-D) in the H.263 video coding framework. Based on the analyzed data, operational C-D curves are obtained through an exhaustive search, and the Lagrangian multiplier method. The proposed scheme has a deterministic feature that satisfies the given computational constraint, regardless of the changing properties of the input video sequence. It is shown that, in terms of PSNR, an optimally chosen operational mode makes a significant difference compared to non-optimal modes. Furthermore, an adaptive scheme iteratively controlling the optimal coding mode is introduced and compared with the fixed scheme, whose operating mode is determined based on the rate distortion model parameters obtained by pre-processing off-line.

To evaluate the performance of proposed scheme according to input video sequences, we apply video sequences other than those involved in the process of model parameter estimation, and show that the model parameters are accurate enough to be applied, regardless of the type of input video sequences. Experimental results demonstrate that, in the adaptive approach, computation reductions of up to 19% are obtained in test video sequences compared to the fixed, while the degradations of the reconstructed video are less than 0.05dB. In addition, the adaptive approach is proven to be more effective with active video sequences than with silent video sequences.

TABLE OF CONTENTS

TABLE OF CONTENTS	iii
LIST OF TABLES	v
LIST OF FIGURES	vii
GLOSSARY	ix
ACKNOWLEDGMENTS	x
DEDICATION	xi
1..INTRODUCTION	1
1.1 MOTIVATION.....	1
1.2 PROBLEM FORMULATION	2
1.3 GENERAL CONTRIBUTIONS	4
1.4 OVERVIEW	5
1.5 SUMMARY.....	6
2..BACKGROUND	7
2.1 GENERIC VIDEO CODER.....	7
2.2 COMPLEXITY ANALYSIS	15
2.3 RATE DISTORTION THEORY	17
2.4 OPTIMIZATION METHODS	19
2.5 SUMMARY.....	22
3..MODEL BASED SUB-PIXEL ACCURACY MOTION ESTIMATION	23
3.1 INTRODUCTION	24
3.2 COMPUTATIONAL COMPLEXITY.....	26
3.3 THE LINEAR CRITERION MODEL	28
3.4 EXPERIMENTAL RESULTS.....	39

3.5	SUMMARY.....	44
4.	REGRESSIVE MODEL BASED RATE DISTORTION OPTIMIZATION	46
4.1	INTRODUCTION	47
4.2	PROBLEM FORMULATION	49
4.3	RATE DISTORTION FUNCTION MODELING	53
4.4	COMPLEXITY ISSUES	56
4.5	EXPERIMENTAL RESULTS	62
4.6	SUMMARY.....	76
5.	DISTORTION AND COMPLEXITY OPTIMIZATION IN SCALEABLE VIDEO CODING SYSTEM	78
5.1	INTRODUCTION	78
5.2	GENERAL PROBLEM FORMULATION.....	80
5.3	COMPLEXITY AND DISTORTION ANALYSIS.....	83
5.4	EXPERIMENTAL RESULTS	94
5.5	SUMMARY.....	104
6.	CONCLUSION	106
	BIBLIOGRAPHY	109
	PARTIAL COPYRIGHT LICENSE	

LIST OF TABLES

Table 3.1	Look-up table for updating motion vectors at half-pel accuracy.....	30
Table 3.2	Look-up table for updating motion vectors at quarter-pel accuracy.....	32
Table 3.3	Evaluation of the proposed method in terms of rate and distortion.....	36
Table 3.4	Evaluation of the proposed method in terms of rate and distortion.....	40
Table 3.5	Performance evaluation in terms of bit rate using test sequences with QP=10	42
Table 3.6	Performance evaluation in terms of bit rate using test sequences with QP=30	42
Table 4.1	Computational complexity for the model-based, RD optimal and TMN5 with the motion vector search range (-15, 15)	60
Table 4.2	Relative rate and distortion model error in RMSE using different averaging window size of the regression model, with the video sequence <i>Miss-America</i>	63
Table 4.3	Rate constrained motion estimation in terms of average rate [bits/frame] and PSNR, QP=15, frames = 50, 10fps	66
Table 4.4	Performance comparisons in terms of PSNR according to the different averaging window size using <i>Miss-America</i> and <i>Carphone</i> sequences.....	71
Table 4.5	Rate distortion performance using the sequence <i>Miss-America</i>	72
Table 4.6	Rate distortion performance using the sequence <i>Carphone</i>	73
Table 5.1	Computational complexity as a function of the search window size for the ME search used	85
Table 5.2	Computation complexity as a function of pruning for the DCT module.....	89
Table 5.3	Average PSNR data and computational complexity of all operation modes, where five video sequences were applied and their results were averaged	92
Table 5.4	Optimal operation modes found through the Lagrangian Method, where the given computational complexity is controlled by the Lagrangian multiplier λ over C-D data	94

Table 5.5	Performance comparison between the fixed and the adaptive control of the operating point (s_1, s_2, s_3) , with video sequences used in the model estimation.....	99
Table 5.6	Performance comparison between the fixed and the adaptive control in the operating point (s_1, s_2, s_3) , with other video sequences not used in the model estimation.....	103

LIST OF FIGURES

Figure 2.1	A generic structure of video coding systems	8
Figure 2.2	The macroblock in the current and previous frame, and the search window.....	9
Figure 2.3	Huffman code for six symbols	13
Figure 2.4	Operation rate distortion function.....	17
Figure 2.5	Convex hull in rate distortion space defined by the Lagrangian multiplier method.....	21
Figure 3.1	Bi-linear interpolation.....	25
Figure 3.2	Characteristic function $f(k)$	29
Figure 3.3	Characteristic function $B(f(k))$	33
Figure 3.4	Description of the gradient based method.....	37
Figure 3.5	Graphical representation of the gradient	38
Figure 3.6	Accuracy of error criterion model.....	41
Figure 3.7	Performance in relative increase of bit rate compared to the full search (%).....	43
Figure 4.1	Block diagram of rate-distortion optimization based on adaptive model.....	51
Figure 4.2	Rate function approximated by the 2 nd order regressive model for the first five frames in the sequence <i>Miss- America</i>	61
Figure 4.3	Distortion function approximated by the 2 nd order regressive model for the first five frames in the sequence <i>Miss-America</i>	62
Figure 4.4	Actual and predicted distortion (a) and rate (b), based on regressive model with the averaging window size 10, and with the video sequence <i>Miss-America</i>	65
Figure 4.5	Comparison of motion vector field between rate-constrained (a) and exhaustive full search (b) motion estimation methods with the frame number =10, QP=15, the video sequence <i>Carphone</i>	68

Figure 4.6	PSNR performance and MV bit-rates according to the given rate constraints 0 to 100, with $QP = 15$, 50 total frames, and the video sequence <i>Carphone</i>	70
Figure 4.7	PSNR performance of rate distortion model with <i>Miss-America</i> sequence	74
Figure 4.8	PSNR performance of rate distortion model with <i>Carphone</i> sequence	76
Figure 5.1	Configurable coding scheme with scalable coding parameters	80
Figure 5.2	Search points according to the different search window in the Three-Step Search.....	83
Figure 5.3	AAN forward DCT flow chart where DCT pruning for $y(0)$ coefficient is represented by the dotted line	88
Figure 5.4	Reconstructed video frames with DCT coefficient pruning ($QP=13$, Intra I-frame, and H.263).....	90
Figure 5.5	Optimal operating modes found through exhaustive search over the real-measured C-D (PSNR) data with test video sequences	96
Figure 5.6	Comparison in subjective quality for two modes, A and B of Figure 5.5 requiring similar computational complexity: the 6 th frame, Inter coding, and $QP=13$ in the sequence <i>Carphone</i>	97
Figure 5.7	Operating mode found by adaptive C-D control in the sequence <i>Forman</i>	102

GLOSSARY

B-Picture	Bi-directionally predicted Picture
C-D	Complexity Distortion
DCT	Discrete Cosine Transform
DP	Dynamic Programming
DPCM	Differential Pulse Coded Modulation
JPEG	Joint Photographic Experts Group
HVS	Human Visual System
H.263	ITU-T international video coding standards for motion picture
I-Picture	Intra coded Picture
MPEG	ISO/IEC international video coding standards for motion picture
PSNR	Peak Signal to Noise Ratio
P-Picture	Predicted (Inter coded) Picture
R-D	Rate Distortion

ACKNOWLEDGMENTS

I would like to thank my supervisors, Dr. Peter F. Driessen and Dr. Pan Agathoklis, of the Department of Electrical and Computer Engineering at the University of Victoria, for their academic support and their patience during the period of this dissertation.

Special thanks are due to Garry Robb, president of AVT Audio Visual Telecommunications Corporation, for sponsoring SCBC Great Awards Scholarships and for supporting my research work.

I would like to thank Dr. R. N. Horspool, Dr. R. L. Kirlin, Dr. A. Basso, and Dr. H. Kalva for their technical comments and suggestions during my oral examination.

I gratefully acknowledge advice, comments, and technical discussions with Mr. Hyunho Jeon, Mr. Chengdong Zhang, and Mr. Thomas R. Huitika.

DEDICATION

To my lovely family

Eunyeong, Oyoon, and Suemin

Chapter 1

INTRODUCTION

1.1 MOTIVATION

Multimedia communications involving video, audio and data has been an interesting topic for researchers as well as industry. Recently, digital video communications in particular has attracted a lot of attention. In the past, in contrast to analog video, digital video required large amounts of storage and computation power, and was prohibitively expensive for users. This was a major reason for digital video being used in specialized areas only. However, the recent advancement of VLSI semiconductor technology has contributed to the emerging digital multimedia world, and enabled wide digital video applications in the real multimedia life, including desktop computer, DVD, interactive video, HDTV and so on.

Another technology, which has brought about revolutionary multimedia development, is video compression technology, based on both data compression and information theory. Physical networks, such as the public switching telephone networks (PSTN), accessible at home, were originally designed to transmit analog speech signals and were not intended for multimedia application. The fastest speed available through the PSTN is 56kbits/sec, which is considered the upper limit for voice modems. This maximum speed is much less than the required bandwidth needed to transmit uncompressed video sequences. For example, assuming that a QCIF video format is transmitted as an uncompressed video sequence, it requires larger bandwidth than 10Mbits/sec, assuming that the frame rate is 30 frames/sec. This feature shows how significant video compression technology is for video transmission, especially over a narrowband network.

It becomes possible to transmit video sequences over a narrow channel bandwidth due to video compression technology, although it requires computational power for the encoding and decoding of the video sequences. Video compression schemes are attractive since they can achieve such a high-compression performance.

1.2 PROBLEM FORMULATION

Image compression makes use of spatial correlation among neighboring pixels in the image frame, and achieves high compression by removing the redundant information contained in the spatial domain. Video compression, however, is different from image compression as it utilizes not only spatial correlation in the same frame but also temporal correlation contained between succeeding image frames. Its current frame is predicted from the previously decoded reference frame, based on estimated motion information. Most video compression standards such as ISO/IEC MPEG1, 2 and MPEG-4 and ITU-T H.261 and H.263 use the motion estimation and compensation technique to achieve a high compression ratio, where each frame is divided into macro blocks, i.e. 16×16 , and its motion vector is searched within the predefined search windows, based on a block motion model. Basically, it assumes that all pixels in the block move in the same direction. The block motion model is widely used for real video coding application because of its efficiency with relatively simple computational complexity. Based on the estimated motion information, the current frame can be predicted from the previously reconstructed frame, and the residual error between the current frame and the predicted frame can be generated. Instead of whole image frame data, residual error and motion information can be transmitted to the decoder, such that a high compression of video coding can be achieved.

Block motion estimation algorithms are categorized based on search strategies, full search and fast search. The full search method, which is also called an exhaustive search, computes cost measures at all possible candidate pixel locations in order to find the motion vector of a macro block. From the control flow and implementation point of view,

it is simple in complexity. However, it requires extensive computation to search the entire search area, which prevents the full search motion estimation algorithm from being implemented on a general purpose computer, and makes it unsuitable for real-time application without embedded special hardware. Hence, many fast-search algorithms, which speed-up by reducing the number of search-pixel locations, have been proposed. Fast-search algorithms can improve video coding speed and made video coding systems suitable for real time implementation. However, they can more easily become trapped at the local minima point rather than at the globally minimum point. Real-time applications, which demand fast and efficient methods, require not only reduced computation cost for searching motion vectors but also lower probability of the algorithm being trapped at a local minimum point.

A fundamental problem of motion compensated video coding is the bit allocation between motion information and residual error from the predicted frame. This is a constrained optimization problem, which needs to be solved from the rate distortion point of view. In fact, an optimal rate-distortion optimization algorithm requires excessive computation because it performs DCT and scalar quantization operation for each candidate motion vector and quantization parameter. In the past, many people carried out research in order to reduce the computational complexity of rate distortion optimized algorithms, that is i.e., interpolation technique and table look-up method. A fast and efficient rate distortion optimization algorithm updating model parameter dynamically within the predefined frame window has been developed by many researchers. It makes rate distortion optimization algorithms viable in real-time applications by reducing the excessive computation complexity associated with motion vector decision, DCT, and quantization operations. On the other hand, the proposed fast and efficient motion estimation algorithms are merged into the rate distortion framework, where the algorithms can contribute to reducing the required computation by pruning out the candidate motion vector to be considered in rate distortion optimization.

Under the computing power constrained environments, scaleable video coding schemes are required, where optimally selecting coding parameters significantly affect overall system performance, with respect to both subject and objective quality. A fundamental problem is that of optimally computing a resource allocation among encoding modules under given constraints, such that the system can make the best usage of limited computing resources to maximize its coding performance in terms of its video quality. We derive a general formulation for the optimization problem through a tradeoff between complexity and distortion in a generic video coding system. Then, we present optimal solutions by way of a fast approximate optimization method, as well as through an exhaustive search method. The proposed method addresses an optimization problem to search for the smallest distortion with the given frame-bit budget, which is based on the Lagrangian relaxation and dynamic programming approach.

1.3 GENERAL CONTRIBUTIONS

The major areas of research interest are efficient motion video coding algorithms and their performance optimization in regard to real-time applications. The specific research area and general contributions are summarized as below.

Fast and efficient motion-estimation algorithm development [70], which guides a trade-off method between computation complexity and accuracy performance, based on general investigations of local minima problem common in block-based fast motion estimation methods.

The development of a fast half-pel search method [69, 72], which significantly affects overall system performance in terms of computational complexity, since the relative importance of half-pel search algorithm in video coding system is comparable to that of the integer-pel fast search method.

Efficient error-concealment techniques [71] are introduced in a low bit rate video coding framework, and real-time applications in video transmission over narrow band networks are taken into account.

The development of an adaptive model-based rate distortion optimization algorithm, which reduces the extensive computation requirements in conventional rate distortion approaches.

An optimally scaleable video coding algorithm [65] is developed, which addresses an optimal resource allocation problem under constraint conditions, the extraction of optimal coding parameters, and a deterministic control scheme.

1.4 OVERVIEW

In this section, chapters 2, 3, and 4 are surveyed. In chapter 2, a basic knowledge of video coding algorithms is introduced, as well as the theoretical background of the proposed algorithms. Generic video coding systems are reviewed by identifying major coding components. A complexity metric is defined, which is used for computational complexity analysis in chapter 5. Rate distortion theory and operational rate distortion theory, which respectively derive an upper bound of performance in given information sources and a specific system, are described and compared. The Lagrangian optimization method and the Dynamic Programming method, which are well known in video coding, are reviewed and compared. In chapter 3, fast and efficient techniques applicable to motion video coding systems are developed; these involve an efficient motion estimation algorithm under consideration of a trade-off between complexity and accuracy performance [70], fast half-pel search methods [69, 72], and an efficient error concealment method [71]. Only a half-pel search method [72] is presented in this chapter due to limited space. In chapter 4, we introduce a rate distortion optimization technique that is based on an adaptive rate distortion model and which subsequently reduces prohibitively extensive computations compared to traditional approaches. An optimally scalable video coding

system is proposed in chapter 5, which gives the best selection of video coding parameters under given computational constraints. The proposed system ensures its deterministic response in complexity performance, that a key feature is demanded in most portable and handheld devices. In chapter 6, we summarize the proposed algorithms and experimental results obtained through our research, and point out areas for future research.

1.5 SUMMARY

In this chapter, motivations and increasing demands in video coding area were introduced, along with growing multimedia markets. Fundamental problems occurring in real video applications of video were identified, and some basic approaches to solving those problems were described. General contributions that were made through research conducted were listed, and an overview of the following chapters was presented.

Chapter 2

BACKGROUND

2.1 GENERIC VIDEO CODER

The generic structure of a video coding system, which is commonly applicable to most international standards such as H.261[37], H.263[8], MPEG1[38], MPEG2[39], and MPEG4[40], is briefly introduced[36, 14, 18]. Figure 2.1 shows a generic video coder, with major coding components that consist of motion estimation, DCT/IDCT, quantizer/inverse quantizer, variable length coder, and so on.

Motion Estimation and Compensation

In motion video sequences, most parts of the pictures change little in successive video frames. Therefore, by sending only the difference between two successive frames video data can be reduced significantly. In other words, it is by temporal redundancy reduction that a video coding system can achieve high compression performance, compared to still-image coding. Temporal redundancy can be further reduced by applying motion compensation techniques in predicting the current picture from the reference picture, although this involves a computationally intensive motion-estimation procedure.

Motion-estimation algorithms can be divided into the following categories, according to their characteristics: block-matching method, pel-recursive, gradient techniques, and transform-domain techniques. The block-matching method is the most practical technique, and is used in most video coding standards because it has a very good search performance when its computational complexity is taken into account.

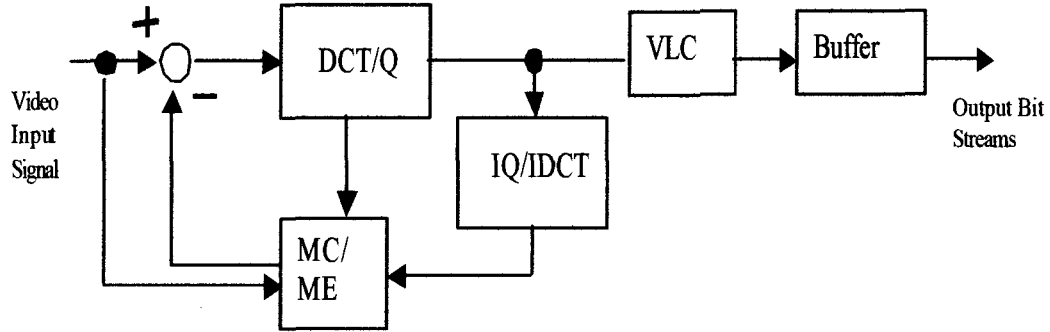


Figure 2.1 A generic structure of video coding systems

In the block matching method, a frame is divided into macroblocks of $N \times N$ (e.g., in most standard codecs, $N = 16$). The best matching macroblock is searched in the given search area. Generally, the search area is a square window of width $(N + 2w)$, where w is the search distance. The decision of the best macroblock match is based on the given cost function. In most video coders, mean absolute error (MAE) and mean squared error (MSE) are commonly used, although MAE is preferred because of its lesser complexity. MAE and MSE are defined as below.

$$MAE(dx, dy) = \frac{1}{N \times N} \sum_{i=-\frac{n}{2}}^{\frac{n}{2}} \sum_{j=-\frac{n}{2}}^{\frac{n}{2}} |F(i, j) - G(i + dx, j + dy)| \quad (2.1)$$

$$MSE(dx, dy) = \frac{1}{N \times N} \sum_{i=-\frac{n}{2}}^{\frac{n}{2}} \sum_{j=-\frac{n}{2}}^{\frac{n}{2}} [F(i, j) - G(i + dx, j + dy)]^2 \quad (2.2)$$

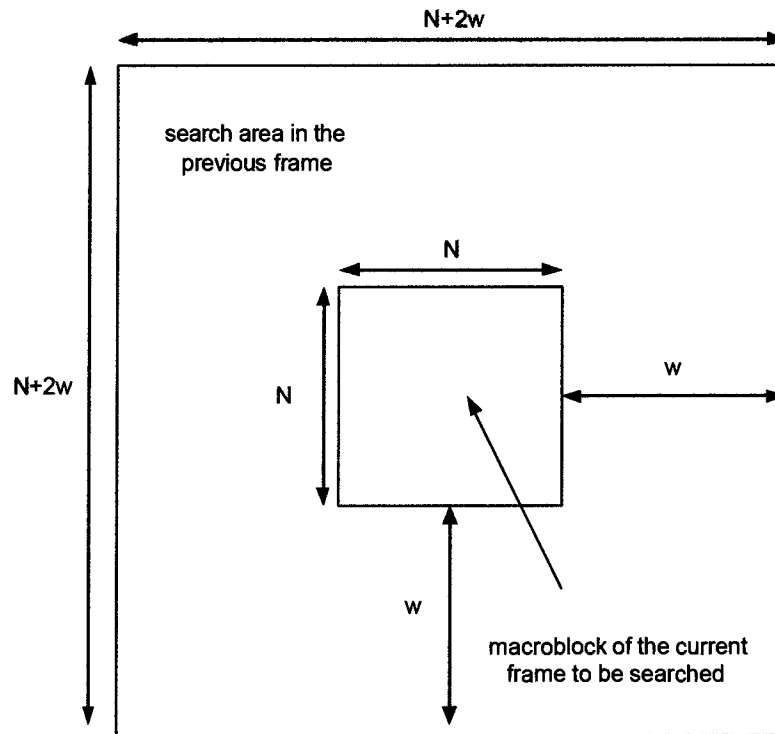


Figure 2.2 The macroblock in the current and previous frame, and the search window

where $F(i, j)$ is the $(N \times N)$ macroblock in the current frame, $G(i, j)$ is the reference $(N \times N)$ macroblock, and (dx, dy) is the search location motion vector.

In regard to the computational requirement of the motion estimation algorithm, the number of search locations and the cost function affect its major complexity. For the exhaustive search, the number of search locations is $(2w+1)^2$, and the MAE cost function requires $2N^2$ arithmetic operations, including addition and subtraction. In the required computation, it is prohibitively intensive to implement such a motion estimation algorithm, especially on general-purpose computers. Therefore, many fast motion

estimation algorithms such as Three Step Search(TSS)[10], 2D LOGarithmic search(2D LOG)[9], and Diamond Search(DS)[11, 12] were developed to reduce this computational complexity.

DCT and IDCT

The spatial redundancy existing between pixels in the picture can be reduced through transform domain coding. After converting the pixels of the time domain into the transform coefficients, most of the energy is concentrated into low frequency coefficients. In other words, it is because of the energy compaction property that the transform domain coding techniques can achieve such a high performance in image data compression. Generally, transform coding is followed by the quantization process, where the transform coefficients are quantized into discrete numbers. In fact, actual data compression is achieved in the quantization process, since most high frequency coefficients are insignificant or zero, and are discarded by the given quantizer.

The energy compaction property affects the compression performance of the transform coding method. From many transform coding methods, Discrete Cosine Transform(DCT) is most often used in compression algorithms, since its rate distortion performance is close to that of the Karhunen-Loeve Transform(KLT), which is known to be optimal. Furthermore, many fast and efficient algorithms for DCT are available, while the KLT transform is too complex to be considered in a real-time implementation. The basic computation in the DCT-based compression system is the transformation of an 8x8 2D image block, which is described as follows.

$$y(k,l) = \frac{c(k)c(l)}{4} \sum_{i=0}^7 \sum_{j=0}^7 x(i,j) \cos \frac{(2i+1)k\pi}{16} \cos \frac{(2j+1)l\pi}{16}, k, l = 0, \dots, 7 \quad (2.3)$$

where $c(k) = \frac{1}{\sqrt{2}}$ for $k=0$ and $c(k) = 1$ otherwise.

The 2-D DCT transform can be decomposed into two 1-D 8-points transforms, and the above equation can be modified as

$$y(k,l) = \frac{c(k)}{2} \sum_{i=0}^7 \left[\frac{c(k)}{2} \sum_{j=0}^7 x(i,j) \cos \frac{(2j+1)k\pi}{16} \right] \cos \frac{(2i+1)l\pi}{16}, k,l = 0,\dots,7 \quad (2.4)$$

where $[\cdot]$ denotes the 1-D DCT of the rows of input $x(i,j)$, which is rewritten below.

$$z_{ii} = \frac{c(l)}{2} \sum_{j=0}^7 x(i,j) \cos \frac{(2j+1)l\pi}{16}, i = 0,\dots,7 \quad (2.5)$$

This row-column decomposition gives a reduction of the required computation that is four times less than that of the direct computation. The 2-D DCT computation requires 4096 multiplications and additions for each. However, by using the row-column decomposition approach, it can be reduced to 1024 multiplications and additions, which is four times less than that of the direct calculation. Although the separability property of DCT has reduced the computational complexity, these numbers are still prohibitive for real-time application. Therefore, many fast DCT computation algorithms have been developed to reduce such a huge computational burden [21].

Quantizer and Inverse Quantizer

Quantization block is one of coding components which yields actual compression through video coding blocks, since the DCT transformation itself does not give any bit rate reduction. Compression gain is controlled in the change of quantization step size. The coarse quantizer gives higher compression, although the picture quality deteriorates. Most video codecs adopt the Uniform Threshold Quantizer(UTQ) where the quantization step size is equal through the whole range of quantized coefficient. According to picture types, there is little difference in quantizing coefficients. Typically, the DC coefficient of

the intra block is divided by the quantizer, with a rounding towards to the nearest integer, while the AC and DC of the inter block are divided by the quantizer, with truncation towards zero. Quantization and inverse quantization in both cases are represented as follows.

For intra DC coefficient,

$$L(x, y) = \frac{C(x, y) \pm q}{2q} \quad (2.6)$$

$$C(x, y) = L(x, y) \times 2q$$

And for inter AC and DC coefficients,

$$L(x, y) = \frac{C(x, y)}{2q} \quad (2.7)$$

$$C(x, y) = \{2L(x, y) \pm 1\} \times q$$

where q , $L(\cdot)$ and $C(\cdot)$ are quantizer, quantization index, and reconstructed coefficient, respectively. The range of quantizer value is from 1 to 31, and the quantized coefficients can be from -2047 to +2047.

Variable Length Coder

The variable length coder (VLC) is one of the coding modules which make the video coding system achieve actual compression, as does the quantization module. The DCT coefficient, the motion vector, and the macro block type information are coded by the VLC in most video coding systems.

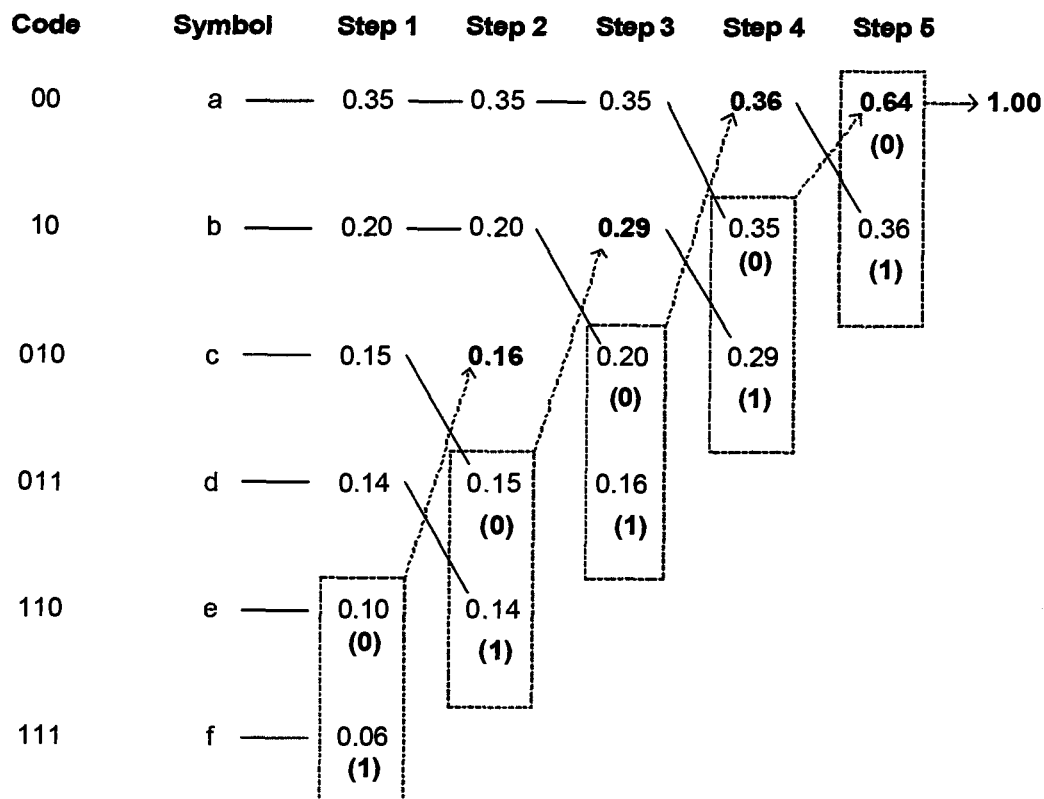


Figure 2.3 Huffman code for six symbols

In regard to the VLC, the code length is varied inversely with the occurrence probability of each symbol. In other words, highly probable symbols are given short codes words, and the less probable symbols are given long code words, respectively.

Two types of VLC, Huffman coding and Arithmetic coding, are commonly used in most video coding systems, while Arithmetic coding is preferred as more compression is demanded. In fact, Huffman coding can not achieve a compression performance as low as the entropy of the encoded symbols, since the symbols are represented in the integral number of bits. However, arithmetic coding can achieve its compression performance

close to the entropy of the coded symbols, since the symbols are coded by a fractional number. A general procedure to generate the Huffman code from the symbols and probability data is described as follows:

Step 1: First, rank all the symbols in the descending order of their probabilities

Step 2: Merge the least two probabilities and reorder them with the merged probability, and continue this merging procedure until it reaches the top node with the probability “1”

Step 3: Assign “0” and “1” to each branch of the combined node. The code word corresponding to each symbol is obtained by reading from the top node to the beginning

An example of Huffman coding is shown in Figure 2.3, where all symbols are variable-length coded, based on the given probabilities. The average bit per symbol in the Huffman code is calculated and compared to the entropy below.

$$E_{avg} = \sum_{i=1}^n b_i p_i = 0.35 \times 2 + 0.20 \times 2 + 0.15 \times 3 + 0.14 \times 3 + 0.10 \times 3 + 0.06 \times 3 = 1.6482$$

And the entropy for all the symbols is given as

$$\begin{aligned} E_{min} &= - \sum_{i=1}^n p_i \log_2 p_i \\ &= -(0.35 \log_2 0.35 + 0.2 \log_2 0.2 + 0.15 \log_2 0.15 + 0.14 \log_2 0.14 + 0.10 \log_2 0.10 + 0.06 \log_2 0.06) \\ &= 2.45bits \end{aligned}$$

The average bits of the Huffman code are not as low as the entropy of the symbols, since each symbol in the Huffman code is represented by the integral number of bits. However,

arithmetic coding can achieve the theoretical entropy, since data consisting of a sequence of symbols are represented in a fractional number [36].

2.2 COMPLEXITY ANALYSIS

When computation power of a specific algorithm on the target processor is estimated, it is more accurate when memory access as well as arithmetic computations is taken into account. A generic complexity metric is defined, based on instruction level analysis [19].

According to attributes, major complexity parameters for implementing application programs on the processor can be divided into three groups, such as memory, computation and control. In regard to memory, bandwidth, size and granularity are dominant factors in deciding implementation complexity. Arithmetic computation related cost is proportional to arithmetic operation type (e.g., addition, division), operation data type (e.g., integer, float), and operation word length (e.g., 1, 2, 4 byte). In control cost, the branch type (e.g., conditional/unconditional, regular/irregular) and its numbers in the program affect overall implementation complexity. Furthermore, memory access pattern, parallelism, and real-time implementation can be taken into account.

However, in this section, RISC-like operations are considered for complexity analysis. They are divided into three categories; arithmetic (e.g., multiplications, additions, subtractions, shift operations, divisions), memory access (e.g., load, store), and control (e.g., if, if then else).

7..COMPLEXITY METRIC

To compare algorithmic complexity, a complexity metric is defined, which is adopted through all the complexity analysis that follows. Complexity metric T , given as the sum of weighted instructions, is represented as

$$T = W_{arith} \cdot N_{arith} + W_{control} \cdot N_{control} + W_{memory} \cdot N_{memory} \quad (2.8)$$

where $N_{arith} = [n_1, n_2, n_3, \dots, n_{ka}]^T$, $N_{control} = [n_1, n_2, n_3, \dots, n_{kc}]^T$, $N_{memory} = [n_1, n_2, n_3, \dots, n_{km}]^T$ are vectors for the number of instructions for arithmetic, control and memory access, and $W_{arith} = [w_1, w_2, w_3, \dots, w_{ka}]^T$, $W_{control} = [w_1, w_2, w_3, \dots, w_{kc}]^T$, $W_{memory} = [w_1, w_2, w_3, \dots, w_{km}]^T$, respectively, their weighting value, which depends on the target application with a particular processor, and ka, kc, km , respectively, the number of instructions. Note that all RISC-like operations are set to one for the sake of simplification, since no particular processor is considered for following complexity analysis.

Computation Power Estimation

To estimate accurate power requirements of an application algorithm on the target processor, power analysis tools as well as knowledge of the target processor architecture are required, which can be too complex and time consuming in real application. Therefore, it is more realistic to estimate the power consumption of each instruction on the processor. Based on the complexity analysis obtained in the instruction level, the required computing power can be estimated by means of a simplified power model. A simple power model with the same weight for all instructions can be defined as [19]

$$Computing_Power_{estimation} = W_t \cdot N_t \cdot C_0 \cdot V_{dd}^2 \quad (2.9)$$

where $W_t = [w_1, w_2, w_3, \dots, w_k]^T$ and $N_t = [n_1, n_2, n_3, \dots, n_k]^T$ are, respectively, vectors for the weighting values and for the number of executions of each instruction, and k , C_0 and V_{dd}^2 are, respectively, the total number of instructions, the capacitive load, and the supply voltage. Once the required power consumption is done, algorithmic complexity can be scaled to meet the given power constraints.

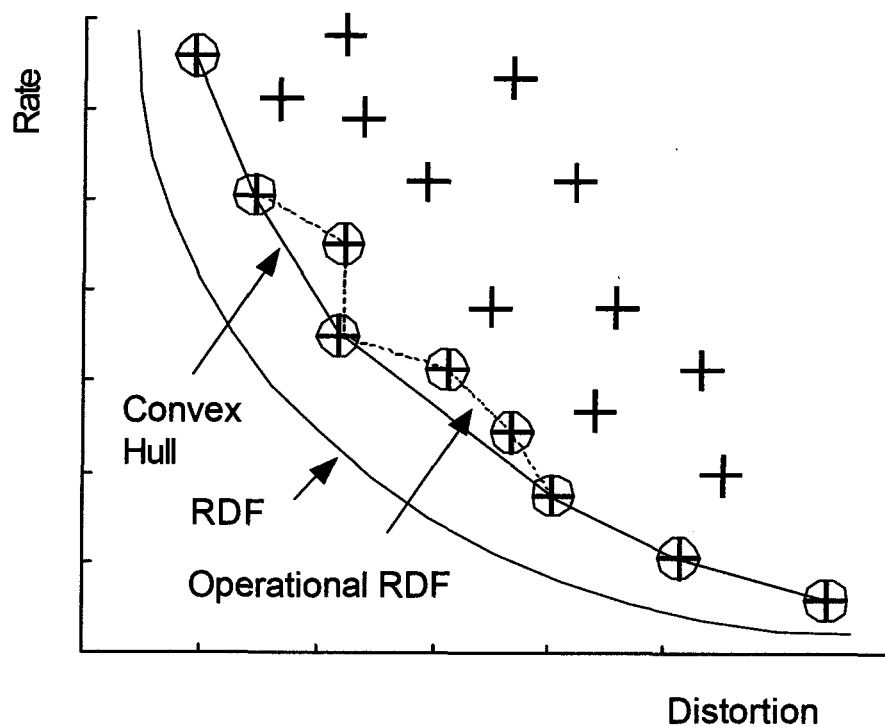


Figure 2.4 Operation rate distortion function

Hence, the complexity analysis and required power estimation of the application program on the target processor is significant, particularly for embedded and portable applications incurring constrained power consumption.

2.3 RATE DISTORTION THEORY

Rate distortion theory, as part of information theory, originates in a paper written by Shannon [35]. It is related to the absolute performance bound of the lossy data compression scheme. Rate distortion function (RDF) is a good tool to describe rate distortion theory, which gives a lower performance bound on the required rate to

represent a source with a given average distortion. In other words, the RDF is concerned with the entropy of a source. In the source coding theorem, the entropy of a source is the minimum rate at which a source can be encoded without information loss. To meet the target rate given its source entropy, a certain information loss is unavoidable. Hence, if a certain maximum rate is given in the system, the minimum average distortion can be derived from the RDF. Conversely the RDF can also be used to find the minimum rate of a data source under a given average distortion. The RDF is continuous, differentiable and non-increasing. Rate distortion theory has significant meaning relevant to the lossy data compression scheme, since its performance bound can be derived from the theorem, while the RDF can be derived explicitly only from simple source models.

Operational Rate Distortion Theory

In every lossy data compression scheme, only a finite set of rate and distortion pairs are available. Operational rate distortion theory (ORDT) is defined in the context of the actual lossy coding scheme, while RDT is continuous and derived from a theoretical source model. Operational rate distortion function (ORDF) consists of a set of rate distortion pairs chosen for optimal performance from all possible discrete rate distortion pairs.

A typical operational rate distortion function is represented in Figure 2.4, where crosses and circles represent all rate distortion pairs, while circles indicate points corresponding to the operational rate distortion curve. A rate distortion pair can belong to an ORDF curve when there is no other rate-distortion point giving a lesser rate for the same distortion. Conversely, it belongs to the ORDF curve if there is no other rate distortion point giving a lesser distortion with the same, or a smaller, rate.

RDT gives the absolute performance bound for a given source regardless of the applied coding scheme, while ORDT derives the optimal performance bound of a given compression scheme. In other words, RDT is used to access the optimal performance of

an actual coding scheme, since it gives the upper bound in the theoretical performance. However, ORDT derives the performance bound of a given coding scheme to achieve its optimal performance. The optimal performance is achieved through optimal bit allocation such that the overall distortion is minimized under the given rate constraint.

Optimal bit allocation means that the available bits are distributed among different sources of information to minimize the resulting distortion. The solution to the bit allocation problem is based on the rate distortion function. Therefore, the optimal bit allocation can be formulated as a constrained optimization problem, and its solution can be found through Lagrangian multiplier method or Dynamic Programming.

2.4 OPTIMIZATION METHODS

Two optimization tools, the Lagrangian multiplier method and Dynamic Programming (DP) [28], are very well known in the area of video compression. In terms of complexity, the Lagrangian multiplier method is usually preferred, although it has the shortcoming of not being able to reach optimal operational points that do not belong to the convex hull. This means the Lagrangian approach does not necessarily provide the overall optimal solutions that are guaranteed in the DP approach.

Lagrangian Multiplier Method

The Lagrangian multiplier method is well known as a mathematical tool for solving constrained optimization problems in a continuous framework. Furthermore, it can also be applied to constrained discrete optimization problems. In fact, a constrained optimization problem for optimal bit allocation is relaxed to an unconstrained problem for dynamic programming. In other words, by applying the Lagrangian multiplier to the hardly constrained problem, the relaxed problem is solved iteratively by searching the Lagrangian multiplier giving the optimal solution. In the context of ORDT, optimization is achieved such that the overall distortion is minimized, subject to the given bit constraints. Basically, it is a constrained problem in the discrete optimization framework.

Note that in the actual video coding system, a finite number of rate distortion points are available. Therefore, the integer version of the Lagrangian multiplier method is described in this section.

Let Q be a member of a finite quantizer set, and $D(Q)$ and $R(Q)$, respectively, its corresponding distortion and rate. Then, the general formulation of the optimal bit allocation problem is defined as follows.

$$\min D(Q), \text{ subject to } R(Q) \leq R_{\max} \quad (2.10)$$

Since the optimization problem is hardly constrained, it is not easy to solve directly. Therefore, the Lagrangian multiplier λ is introduced into the equation so that it can be relaxed to the unconstrained optimization problem, which can be defined as follows.

$$\min D(Q) + \lambda * R(Q) \quad (2.11)$$

where the Lagrangian multiplier λ is non-negative, $\lambda \geq 0$. By searching for an optimal non-negative λ iteratively, the optimal solution to (2.11) can be found. It is also an optimal solution to the constrained problem (2.10). If the rate distortion function is convex and non-increasing, then λ is explained as the derivative of the distortion, with respect to the rate.

$$-\frac{dD}{dR} = \lambda \quad (2.12)$$

Equivalently it can be changed as below with respect to the distortion.

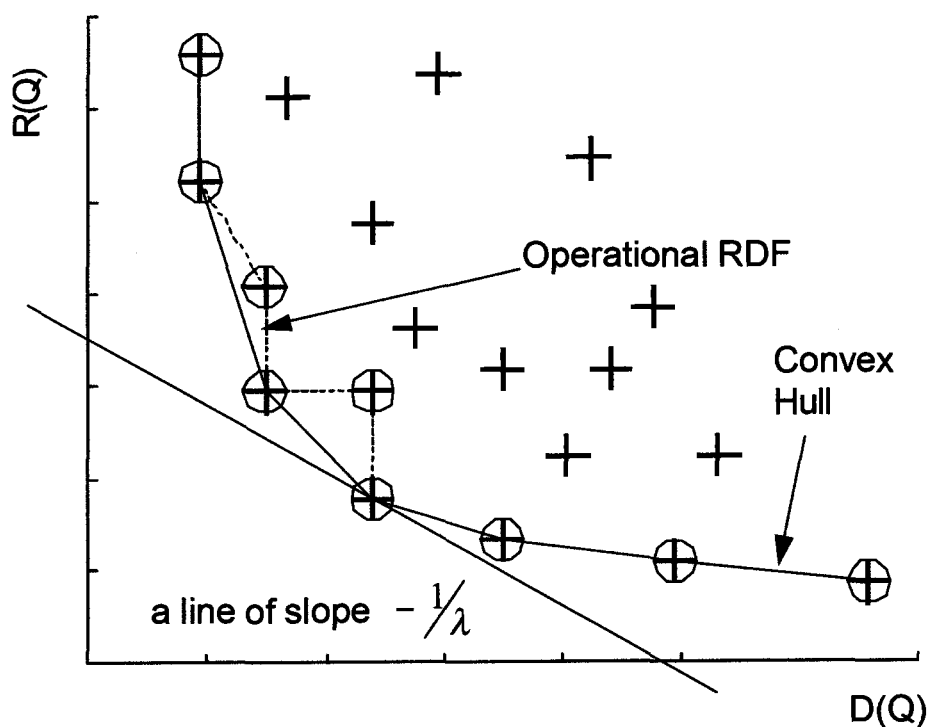


Figure 2.5 Convex hull in rate distortion space defined by the Lagrangian multiplier method

$$-\frac{dR}{dD} = \frac{1}{\lambda} \quad (2.13)$$

Based on these properties of the Lagrangian multiplier λ , fast search methods for optimal λ can be applied [32, 33, 34]. The Lagrangian multiplier method can access only the operational point. It is on the convex hull which consists of optimal operating points connected by straight lines. In fact, the operational rate distortion function is not necessarily convex, while the rate distortion function that is based on rate distortion theory is a non-increasing convex function. In other words, the Lagrangian multiplier of

the unconstrained optimization problem represents the line of slope $-\frac{1}{\lambda}$, which is tangent to the operational rate distortion curve. Therefore, optimal rate distortion points of the Lagrangian multiplier method are found by sweeping the λ from 0 to infinity, which consist of the convex hull being connected by straight lines between points. As is shown in Figure 2.5, all rate distortion points are located above the line defined by the Lagrangian multiplier. It means that any operating point above the convex hull is detected as an optimal solution in the Lagrangian approach.

2.5 SUMMARY

In this chapter, we reviewed topics in which a fundamental knowledge is required in the following chapter. First, a traditional video coding system was introduced, which involves motion estimation and compensation, DCT/IDCT, quantization, variable coding and so on. Then, this system's complexity analysis was described. The rate distortion theory, originating in the information theory, was introduced and compared to the operational rate distortion theory, which can be applied to the actual video coding system. Optimization tools well known in video coding application were introduced and compared with each other; these are the Lagrangian multiplier method and DP.

Chapter 3

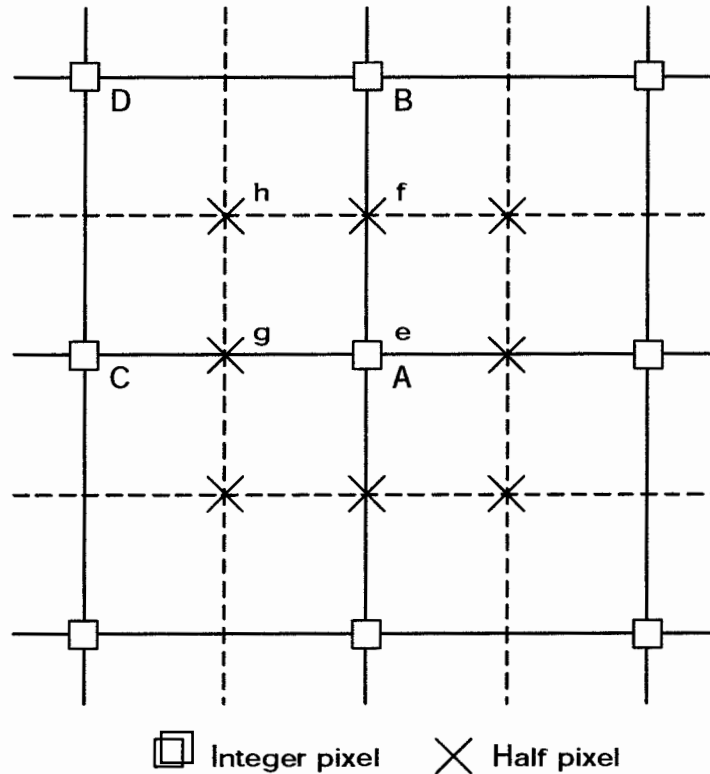
MODEL BASED SUB-PIXEL ACCURACY MOTION ESTIMATION

Sub-pixel accuracy takes up a significant portion of the motion estimation with respect to the computational complexity of video coding. The error criterion function of motion estimation is well represented by a mathematical expression such as quadratic and linear model around the optimal point. Pre-computed error criterion values computed at full-pixel accuracy can be used to derive the motion vector and the error criterion values at sub-pixel accuracy. Based on a linear model function, explicit solutions of the motion vector and the error criterion values at sub-pixel accuracy are derived, which results in the dramatic reduction of computing complexity during the motion estimation process. In addition, a gradient based method is proposed and applied in search of the optimal point which improves further the motion estimation performance while the complexity increase remains negligible. On the other hand, video coding get affected by the accuracy of error criterion model, whose performance changes according to the given coding environment defined by the property of input sequence as well as quantization parameter of coding framework. As a sequel, the maximum coding performance would be achievable if the error criterion model is switched to the one leading to the best performance under a given coding condition. Through experiments carried out in the h.263 framework, it has been proven that the proposed method dynamically switching between two linear and quadratic models can outperform other two methods, while neither of two methods performs the best all the time.

3.1 INTRODUCTION

In motion video coding, only the differences in consecutive frames are encoded to remove temporal redundancy, whereby the high coding performance is achieved. Coding efficiency can be further improved with motion compensated video coding, which needs the motion information of each coding macroblock in the frame. The motion vector information is evaluated at either full-pixel or sub-pixel accuracy. Since more accurate motion estimation leads to better coding performance, motion estimations at sub-pixel accuracy (for example, half-pel, quarter-pel) are desirable and are adopted in the video coding standards. On the other hand, the sub-pixel accuracy mode incurs increased complexity in terms of computation and data transfer.

Motion compensation complexity at sub-pixel accuracy can be reduced using a mathematical model for error criterion such as mean absolute difference (MAD)[66-68]. For instance, the error criterion values at half-pixel accuracy are estimated by interpolating the error criterion values of surrounding full-pixels obtained from the previous explicit computation at full-pixel level. In the same manner, quarter-pixel accuracy can be derived from error criterion values obtained at half-pixel accuracy, and vice versa. Some researchers [66, 67] introduce a linear interpolation model for the error criterion function, where model parameters are defined empirically. In [68], a quadratic approximation model is adopted and explicit solutions for motion vectors are derived, as well as error criterion values. The quadratic approximation model is tractable mathematically, but it does not necessarily lead to a better performance than the linear mode approach [14]. In the paper, we derive explicit solutions for the motion vector, as well as an error criterion with a linear approximation model. It is evident geometrically that the optimal point is located in close proximity to the direction where the gradient between two pixels leads to the maximum. A proposed gradient-based method further improves the motion estimation accuracy, and can be applied to other model-based methods in the same manner. Besides, the motion estimation accuracy can be improved by alternatively switching between two models.



$$e = A$$

$$f = (A + B + 1)/2$$

$$g = (A + C + 1)/2$$

$$h = (A + B + C + D + 2)/4$$

Figure 3.1 Bi-linear interpolation

Following, in section 3.2, the computational complexity of the motion estimation process is addressed in regard to accuracy. A linear error criterion model is introduced and explicit solutions are derived for the optimal motion vector and the error criterion value in section 3.3. In section 3.4, a gradient-based method is introduced and verified through experiments using test sequences. In addition, a switching model based method is introduced and verified through experiments using test sequences, concluding remarks follow in section 3.5.

3.2 COMPUTATIONAL COMPLEXITY

From a practical implementation perspective, the computational complexity of motion estimation is analyzed. Full search motion estimation is computationally too intensive, where the complexity increases quadratically with respect to the sub-pixel accuracy. Ordinarily the multi-step search is adopted with video coding standards. For instance, in case of a two- steps search corresponding to half-pixel accuracy, the first optimal motion vector is searched exhaustively at full-pixel, which is named as the sub-optimal motion vector in the paper. Then, two approaches are possible in obtaining the optimal motion vector at the sub-pixel accuracy. A conventional method which relies on a direct computation of the error criterion function from interpolated pixels data has been used in many real applications. To be more specific the surrounding eight half-pixel locations of the sub-optimal vector are searched for the optimal motion vector. As an example, half-pixel bi-linear interpolation [4] is described in Figure 3.1. In the same way, more accurate vectors, such as at the quarter-pixel accuracy, can be searched, and vice versa. On the other hand, error criterion values are modeled with a mathematical formula and its optimal vector is derived from the model. From computing complexity perspective, two methods are analyzed and compared each other in following.

Let a video frame consist of macroblocks. For the complexity analysis, we assume the following; a frame size of 176×144 QCIF format, a macroblock size of 16×16 , and the MAD as an error criterion. Then, the MAD calculation can be represented as below.

$$MAD(x, y) = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |P(i, j) - R(i + x, j + y)| \quad (3.1)$$

where $P(i, j)$ is the $N \times N$ macroblock being compressed in the present frame; $R(i, j)$ is the reference $N \times N$ macroblock in the previous frame; x and y are the search location motion vectors; N is the macroblock size of 16; i and j are horizontal and vertical

coordinates in the macroblock, respectively. The evaluation of each MAD cost function requires 2×256 load operations, 256 subtraction operations, 1 division operation, 1 store operation, and 1 data compare operation. Then, the complexity of MAD in terms of the number of operations, C_{mad} becomes 1035 operations [14].

When the complexity of a single MAD evaluation is taken into account, as shown above, an exhaustive search requires intensive computing power from a practical implementation perspective. In an effort to reduce the computational complexity, many fast search methods, which have different search patterns and different number of search points, have been heuristically developed as alternative solutions to an exhaustive search. Assuming TSS is adopted as one of the fast full-pixel searches, the overall complexity per macroblock, C_{mb} is derived as the sum of the first and second step and given as

$$C_{mb} = 25 \times C_{mad} + 8 \times \frac{w}{2} \times C_{mad} \quad (3.2)$$

where w represents sub-pixel accuracy (e.g., $w = 2, 4$ for half-pixel and quarter-pixel, respectively). It is noteworthy that the complexity of the second step takes up a larger portion among the overall computing operations as the sub-pixel accuracy increases.

For instance, the portion of the second step is 24%, and 39% at half-pixel and quarter-pixel accuracy, respectively. As described above, the overall complexity of motion estimation is significantly affected by the complexity of the second step in the conventional explicit method. However, the model-based MAD approximation method requires a negligible operation for the second step. In an example method [68], required computing operations involved in the decision process for the optimal motion vector are described, where major computations consist of the comparison operation. Let k_x and k_y denote variables defined as in [68] and computed from the pre-computed neighboring MAD values in horizontal and vertical directions. Then the horizontal and vertical

components of optimal motion vector x^* and y^* are defined from the variables directly. First, the horizontal component x^* is computed as below.

$$x^* = \begin{cases} x_0 + \frac{1}{3}, & \text{when } k_x < \frac{1}{3} \\ x_0, & \text{when } \frac{1}{3} \leq k_x \leq 3 \\ x_0 - \frac{1}{3}, & \text{when } k_x > 3 \end{cases}$$

In the same manner, the vertical components y^* can be calculated. To define each component either horizontal or vertical component as shown in (3.1) and (3.2), 3 comparison operations take places at most. Referring to [68], computing the variables k_x and k_y requires a total 6 (i.e., 2 subtractions and 1 division for each). Consequently the total number of required operations become 12 at most. In addition, it is noteworthy that the computing requirement does not change regardless of the accuracy level of sub-pixel motion estimation while the complexity is dependent on the accuracy level.

3.3 THE LINEAR CRITERION MODEL

Let $\varepsilon(x, y)$ represent the error criterion value between the current block and the reference block at pixel location (x, y) of the search area. Then, the criterion function can be approximated using a symmetric, separable, and linear model given below.

$$\varepsilon(x, y) = b |x - a| + d |y - c| + \varepsilon_{\infty} \quad (3.3)$$

where parameters a and c are the theoretical optimal points and ε_{∞} is the optimal criterion error obtained at infinite resolution.

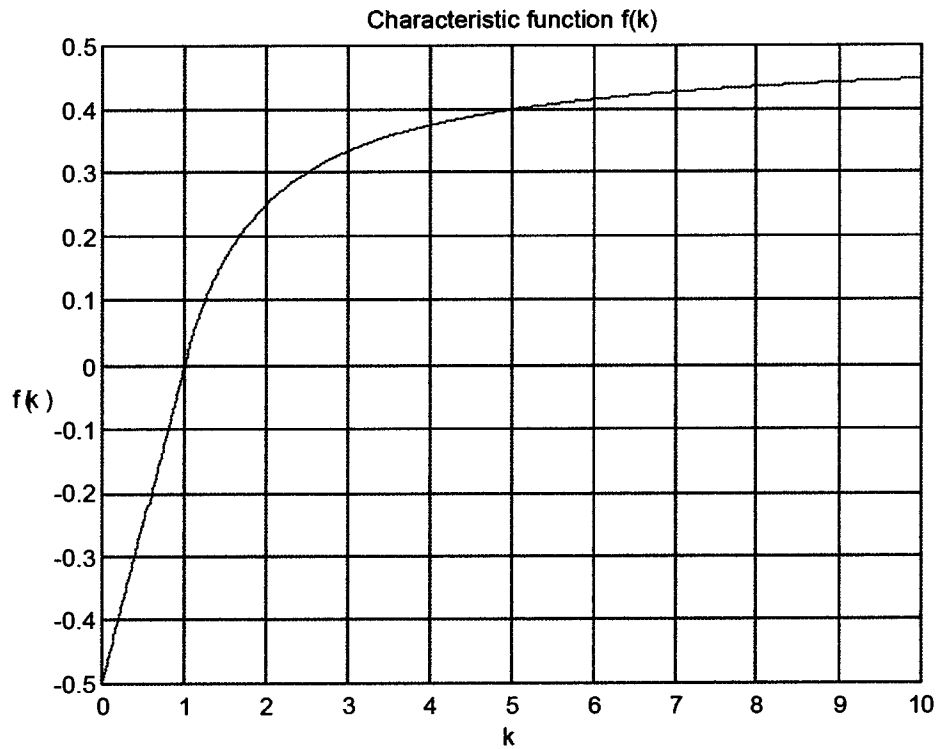


Figure 3.2 Characteristic function $f(k)$

Assume that $\varepsilon(x_0, y_0)$, $\varepsilon(x_0 + 1, y_0)$, $\varepsilon(x_0 - 1, y_0)$, $\varepsilon(x_0, y_0 + 1)$, and $\varepsilon(x_0, y_0 - 1)$ are criterion values at integer-pixel resolution, corresponding to the pixel point (x_0, y_0) and the surrounding points, respectively. As shown in (3.3), the model function is separable for the horizontal and vertical direction.

Hence the model parameters can be computed separately in both directions. First, model parameters a and b are computed using horizontal criterion values as below.

Table 3.1 Look-up table for updating motion vectors at half-pel accuracy

$f(k)$	k	Decision Process
$-\frac{1}{2}$	0	$0 < k \leq \frac{1}{4}, \quad x_0 - a = -\frac{1}{2}$
$-\frac{3}{8}$	$\frac{1}{4}$	
$-\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4} < k \leq \frac{3}{4}, \quad x_0 - a = -\frac{1}{4}$
$-\frac{1}{8}$	$\frac{3}{4}$	
0	1	$\frac{3}{4} < k < \frac{4}{3}, \quad x_0 - a = 0$
$\frac{1}{8}$	$\frac{4}{3}$	
$\frac{1}{4}$	2	$\frac{4}{3} \leq k < 4, \quad x_0 - a = \frac{1}{4}$
$\frac{3}{8}$	4	
$\frac{1}{2}$	∞	$4 \leq k < \infty, \quad x_0 - a = \frac{1}{2}$

$$\varepsilon(x_0, y_0) = b|x_0 - a| + d|y_0 - c| + \varepsilon_\infty$$

$$\varepsilon(x_0 + 1, y_0) = b|x_0 + 1 - a| + d|y_0 - c| + \varepsilon_\infty$$

$$\varepsilon(x_0 - 1, y_0) = b|x_0 - 1 - a| + d|y_0 - c| + \varepsilon_\infty$$

And c and d can be computed using the vertical criterion values in the same manner.

$$\varepsilon(x_0, y_0) = b|x_0 - a| + d|y_0 - c| + \varepsilon_\infty$$

$$\varepsilon(x_0, y_0 + 1) = b|x_0 - a| + d|y_0 + 1 - c| + \varepsilon_\infty$$

$$\varepsilon(x_0, y_0 - 1) = b|x_0 - a| + d|y_0 - 1 - c| + \varepsilon_\infty$$

Then, the optimal criterion error ε_∞ can be computed using the computed parameters and can be written as follows.

$$\varepsilon_\infty = \varepsilon(x_0, y_0) - b|x - a| - d|y - c| \quad (3.4)$$

Explicit solutions to the separable linear model equation (3.3) are derived with respect to the optimal pixel points a , c and optimal error criterion ε_∞ . We begin by computing the criterion differential values and the ratio, k_x at the pixel (x_0, y_0) in the horizontal direction as follows.

$$\varepsilon(x_0 + 1, y_0) - \varepsilon(x_0, y_0) = b(|x_0 + 1 - a| - |x_0 - a|) \quad (3.5)$$

$$\varepsilon(x_0 - 1, y_0) - \varepsilon(x_0, y_0) = b(|x_0 - 1 - a| - |x_0 - a|)$$

$$k_x = \frac{\varepsilon(x_0 + 1, y_0) - \varepsilon(x_0, y_0)}{\varepsilon(x_0 - 1, y_0) - \varepsilon(x_0, y_0)} = \frac{|x_0 + 1 - a| - |x_0 - a|}{|x_0 - 1 - a| - |x_0 - a|} = \frac{|x_0 - a + 1| - |x_0 - a|}{|x_0 - a - 1| - |x_0 - a|}$$

where $|x_0 - a|$ is the horizontal distance of the optimal point from the point (x_0, y_0) with the condition $|x_0 - a| < \frac{1}{2}$. Then $x_0 - a$ can be derived as a linear function of k_x , which is named a decision characteristic function throughout the section.

$$x_0 - a = f(k_x) = \frac{k_x - 1}{2} \quad \text{if } -\frac{1}{2} < (x_0 - a) \leq 0 \quad (3.6)$$

$$x_0 - a = f(k_x) = \frac{k_x - 1}{2k_x} \quad \text{otherwise}$$

Table 3.2 Look-up table for updating motion vectors at quarter-pel accuracy

$f(k)$	k	Decision Process
$-\frac{1}{2}$	0	$0 < k \leq \frac{1}{4}, \quad B = -\frac{3}{8}$
$-\frac{3}{8}$	$\frac{1}{4}$	
$-\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4} < k \leq \frac{3}{4}, \quad B = -\frac{1}{4}$
$-\frac{1}{8}$	$\frac{3}{4}$	
0	1	$\frac{3}{4} < k < \frac{4}{3}, \quad B = 0$
$\frac{1}{8}$	$\frac{4}{3}$	
$\frac{1}{4}$	2	$\frac{4}{3} < k \leq 4, \quad B = \frac{1}{4}$
$\frac{3}{8}$	4	
$\frac{1}{2}$	∞	$4 < k \leq \infty, \quad B = \frac{3}{8}$

Similarly, define k_y in the vertical direction. The vertical distance of the optimal point from the point (x_0, y_0) , $y_0 - c$ is computed using the vertical criterion values and the condition $|y_0 - c| < \frac{1}{2}$, and is given as follows.

$$k_y = \frac{\varepsilon(x_0, y_0 + 1) - \varepsilon(x_0, y_0)}{\varepsilon(x_0, y_0 - 1) - \varepsilon(x_0, y_0)} = \frac{|y_0 + 1 - a| - |y_0 - a|}{|y_0 - 1 - a| - |y_0 - a|} = \frac{|y_0 - a + 1| - |y_0 - a|}{|y_0 - a - 1| - |y_0 - a|} \quad (3.7)$$

$$y_0 - c = f(k_y) = \frac{k_y - 1}{2} \quad \text{if } -\frac{1}{2} < (y_0 - c) \leq 0$$

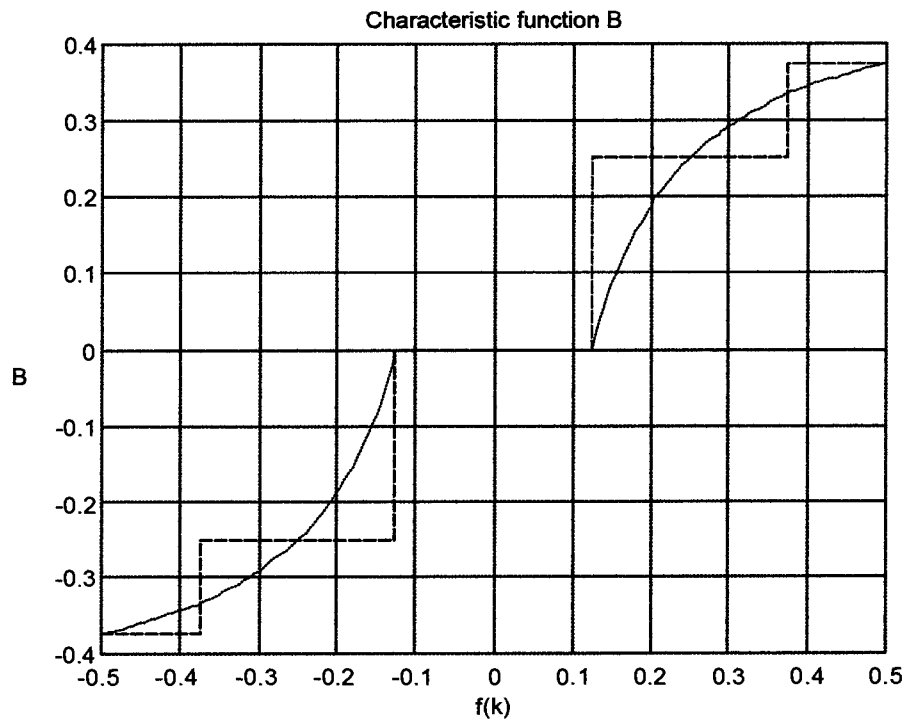


Figure 3.3 Characteristic function $B(f(k))$

$$y_0 - c = f(k_y) = \frac{k_y - 1}{2k_y} \quad \textit{otherwise}$$

Let k denote both k_x and k_y for the purpose of simplicity. The decision characteristic function $f(k)$ is an increasing function of k from $-\frac{1}{2}$ to $\frac{1}{2}$ when k is changing from 0 to infinity as plotted in Figure 3.2. Its look-up table is given in Table 3.1 when half sub-pixel accuracy is assumed. The criterion differences between the two surrounding pixels in the horizontal and vertical directions are given respectively as follows.

$$\varepsilon(x_0 + 1, y_0) - \varepsilon(x_0 - 1, y_0) = b(|x_0 + 1 - a| - |x_0 - 1 - a|)$$

$$\varepsilon(x_0, y_0 + 1) - \varepsilon(x_0, y_0 - 1) = d(|y_0 + 1 - c| - |y_0 - 1 - c|)$$

And the model parameters b and d are written as

$$b = \frac{\varepsilon(x_0 + 1, y_0) - \varepsilon(x_0 - 1, y_0)}{|x_0 - a + 1| - |x_0 - a - 1|}$$

$$d = \frac{\varepsilon(x_0, y_0 + 1) - \varepsilon(x_0, y_0 - 1)}{|y_0 - c + 1| - |y_0 - c - 1|}$$

Substituting the model parameters, the optimal criterion value at infinite resolution e_∞ is written as

$$\begin{aligned} e_\infty &= \varepsilon(x_0, y_0) - b|x - a| - d|y - c| \\ &= \varepsilon(x_0, y_0) - \frac{|x - a|}{|x_0 - a + 1| - |x_0 - a - 1|} \times [\varepsilon(x_0 + 1, y_0) - \varepsilon(x_0 - 1, y_0)] \\ &\quad - \frac{|y - c|}{|y_0 - c + 1| - |y_0 - c - 1|} \times [\varepsilon(x_0, y_0 + 1) - \varepsilon(x_0, y_0 - 1)] \end{aligned} \quad (3.8)$$

Then the criterion value at (x, y) is computed as

$$\begin{aligned}
\varepsilon(x, y) &= \varepsilon_x + b|x-a| + d|y-c| \\
&= (\varepsilon(x_0, y_0) - b|x_0-a| - d|y_0-c|) + b|x-a| + d|y-c| \\
&= \varepsilon(x_0, y_0) - \frac{|x_0-a| - |x-a|}{|x_0-a+1| - |x_0-a-1|} \times [\varepsilon(x_0+1, y_0) - \varepsilon(x_0-1, y_0)] \\
&\quad - \frac{|y_0-c| - |y-c|}{|y_0-c+1| - |y_0-c-1|} \times [\varepsilon(x_0, y_0+1) - \varepsilon(x_0, y_0-1)] \\
&= \varepsilon(x_0, y_0) - B_x [\varepsilon(x_0+1, y_0) - \varepsilon(x_0-1, y_0)] - B_y [\varepsilon(x_0, y_0+1) - \varepsilon(x_0, y_0-1)]
\end{aligned} \tag{3.9}$$

where the parameters B_x and B_y represent $\frac{|x_0-a| - |x-a|}{|x_0-a+1| - |x_0-a-1|}$ and $\frac{|y_0-c| - |y-c|}{|y_0-c+1| - |y_0-c-1|}$ respectively. B_x and B_y can be further reduced, since $|x_0-a|$ and $|y_0-c|$ are less than $\frac{1}{2}$. First, B_x is derived as below

$$B_x = \begin{cases} -\frac{1}{2} - \frac{1}{16f(k_x)}|x-a|, & \text{if } -\frac{1}{2} \leq f(k_x) = (x_0-a) < 0 \\ 0, & \text{if } f(k_x) = (x_0-a) = 0 \\ \frac{1}{2} - \frac{1}{16f(k_x)}|x-a|, & \text{if } 0 < f(k_x) = (x_0-a) \leq \frac{1}{2} \end{cases} \tag{3.10}$$

Similarly the parameter B_y is given as below

$$B_y = \begin{cases} -\frac{1}{2} - \frac{1}{16f(k_y)}|y-c|, & \text{if } -\frac{1}{2} \leq f(k_y) = (y_0-c) < 0 \\ 0, & \text{if } f(k_y) = (y_0-c) = 0 \\ \frac{1}{2} - \frac{1}{16f(k_y)}|y-c|, & \text{if } 0 < f(k_y) = (y_0-c) \leq \frac{1}{2} \end{cases} \tag{3.11}$$

Table 3.3 Evaluation of the proposed method in terms of rate and distortion

Half-pixel ME	<i>Miss-America</i>		<i>Foreman</i>	
	Rate (Kbps)	Distortion (PSNR)	Rate (Kbps)	Distortion (PSNR)
Exhaustive	20.58	36.13	80.42	30.79
Quadratic [68]	21.21	36.01	84.25	30.62
Linear	21.72	35.96	86.58	30.62

As an instance, the motion vector update of half-pixel resolution can be described below. Basically it is not necessary to compute the exact location of the motion vector as far as motion estimation is concerned with determining motion vector in half-pel accuracy. In other words, the point is to find where the minimum point is most closely located among $x, y \in \{-\frac{1}{2}, 0, \frac{1}{2}\}$ locations.

STEP 1: Compute the criterion differential k_x . It can be converted to $f(k_x) = x_0 - a$, which is the location value of the minimum in the horizon direction, and is equivalent to the distance from the origin. In the same manner, k_y and $f(k_y)$ can be driven.

STEP 2: As shown in the table 3.1 and the equation (3.12), the horizontal motion vector in the half-pixel accuracy $x \in \{-\frac{1}{2}, 0, \frac{1}{2}\}$, are determined using the ratio k_x . In the same manner $y \in \{-\frac{1}{2}, 0, \frac{1}{2}\}$ is derived using the equation (3.13).

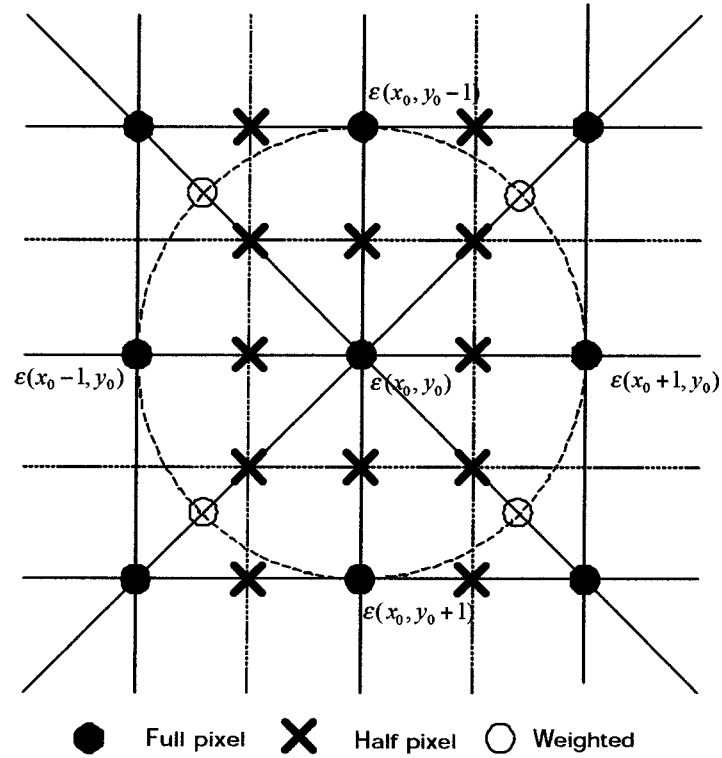


Figure 3.4 Description of the gradient based method

$$x = \begin{cases} x_0 + \frac{1}{2}, & \text{when } k_x < \frac{1}{2} \\ x_0, & \text{when } \frac{1}{2} \leq k_x \leq 2 \\ x_0 - \frac{1}{2}, & \text{when } k_x > 2 \end{cases} \quad (3.12)$$

$$y = \begin{cases} y_0 + \frac{1}{2}, & \text{when } k_y < \frac{1}{2} \\ y_0, & \text{when } \frac{1}{2} \leq k_y \leq 2 \\ y_0 - \frac{1}{2}, & \text{when } k_y > 2 \end{cases} \quad (3.13)$$

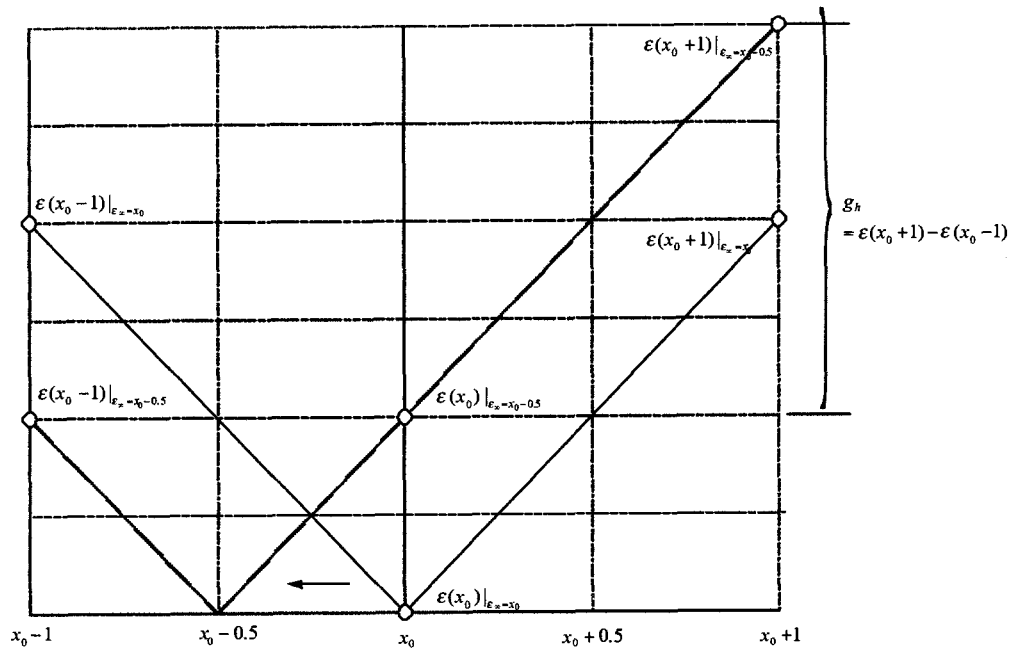


Figure 3.5 Graphical representation of the gradient

In motion video coding such as MPEG, there are certain cases to evaluate the error criterion values for motion vectors. In such cases, the equation (3.14) is used to compute the error criterion for the determined motion vector obtained by (3.12) and (3.13).

$$\epsilon = \begin{cases} \epsilon(x_0, y_0), & \text{when } \frac{1}{2} < k_x < 2, \text{ and } \frac{1}{2} < k_y < 2 \\ \epsilon(x_0, y_0) - \frac{1}{2} \times |\epsilon(x_0+1, y_0) - \epsilon(x_0-1, y_0)|, & \\ \text{when } k_x < \frac{1}{2} \text{ or } k_x > 2, \text{ and } \frac{1}{2} < k_y < 2 \\ \epsilon(x_0, y_0) - \frac{1}{2} \times |\epsilon(x_0, y_0+1) - \epsilon(x_0, y_0-1)|, & \\ \text{when } \frac{1}{2} < k_x < 2 \text{ and } k_y < \frac{1}{2}, \text{ or } k_y > 2 \\ \epsilon(x_0, y_0) - \frac{1}{2} \times |\epsilon(x_0+1, y_0) - \epsilon(x_0-1, y_0)| - \frac{1}{2} |\epsilon(x_0, y_0+1) - \epsilon(x_0, y_0-1)|, & \\ \text{when } (k_x < \frac{1}{2}, \text{ or } k_x > 2) \text{ and } (k_y < \frac{1}{2}, \text{ or } k_y > 2) \end{cases} \quad (3.14)$$

3.4 EXPERIMENTAL RESULTS

Experiments are carried on in H.263 framework. *Miss-America and Foreman* in size of QCIF(176x144) are selected as test sequences. Encoding frame rate are 10 fps achieved by skipping every two frames of the original sequences. For the sake of performance comparison, three search methods are implemented: a conventional exhaustive method, a linear model-based, and a quadratic model-based. They are evaluated in terms of rate and distortion as shown in Table 3.3. The conventional exhaustive method generates optimal data as a reference in the comparison since it directly measures the error criterion values of all surrounding half-pixels. It is clarified in the table where the conventional method represents the best performance in terms of rate and distortion among three methods.

On the other hand, two different model-based methods are compared without a significant difference although the quadratic is slightly better than the linear approach. From the experiments, it is shown that the model-based approaches out-perform the conventional method with regard to computing complexity although it incurs a slight sacrifice of performance in terms of rate and distortion.

Gradient based method

When the optimum point is computed using the approach described above, only pixel points located on the horizontal and the vertical are taken into account. In a proposed gradient-based method, however, it is shown that the decision performance can be improved by considering all 8 surrounding pixels, by including 4 pixels in the diagonal direction. Basically, the gradient value is used to refine the location of point. There are four gradient directions, corresponding to horizontal, vertical, and two diagonals. The gradient can be computed simply by taking the difference between two pixels in one direction, while in case of the diagonals, the gradient should be adjusted for a fair comparison with the other, since its geometrical distance from the center is longer, as shown in Figure 3.4. Then, the gradients can be represented as follows and it is graphically shown in Figure 3.5.

Table 3.4 Evaluation of the proposed method in terms of rate and distortion

Half-pixel ME	<i>Miss-America</i>		<i>Foreman</i>	
	Rate (Kbps)	Distortion (PSNR)	Rate (Kbps)	Distortion (PSNR)
Linear	21.72	35.96	86.58	30.62
Proposed	21.15	35.94	83.33	30.57

$$g_h = \varepsilon(x_0 + 1, y_0) - \varepsilon(x_0 - 1, y_0) \quad (3.15)$$

$$g_v = \varepsilon(x_0, y_0 + 1) - \varepsilon(x_0, y_0 - 1)$$

$$g_{d1} = w \times \{ \varepsilon(x_0 + 1, y_0 + 1) - \varepsilon(x_0 - 1, y_0 - 1) \}$$

$$g_{d2} = w \times \{ \varepsilon(x_0 - 1, y_0 + 1) - \varepsilon(x_0 + 1, y_0 - 1) \}$$

where the parameter w is the weighting factor to adjust values in the diagonal directions. Assuming the same linear model is adopted for the error criterion function, the weighting parameter w can be set to $1/\sqrt{2}$. It is evident that the minimum gradient value among all four gradients represents the overall gradient direction of the error criterion function. As shown in Figure 3.4, the area of optimum point is geometrically placed in the same direction as the minimum gradient. Hence, the optimum point can be computed in the same manner by applying the equations (3.12) and (3.13) to two full pixel points located in the minimum gradient direction. Figure 3.5 shows that the gradient value decreases to the minimum at the center, while it increases as the optimal point moves away from the center.

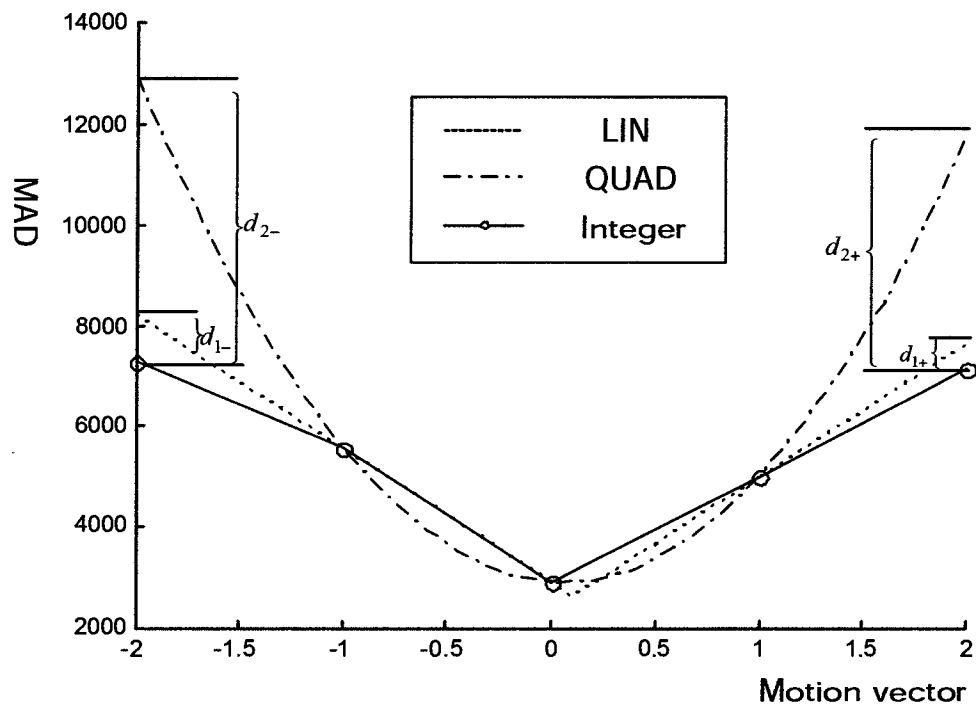


Figure 3.6 Accuracy of error criterion model

The performance of the proposed scheme has been evaluated in terms of bit rate and PSNR, using test video sequences as shown in Table 3.4. Rate saving was obtained up to a maximum of 3%, while the PSNR quality sacrifice was negligible. Experiments have verified that the proposed scheme improves the video coding performance in terms of bit rates, where the gradient was taken into account in a search for the optimal point. Consequently, the proposed scheme is proved for a more accurate performance in motion estimation.

Table 3.5 Performance evaluation in terms of bit rate using test sequences with QP=10

Error Models		Test sequences (rate, %)			
		<i>Foreman</i>	<i>Carphone</i>	<i>Mobile</i>	<i>Container</i>
QP =10	FULL [kbps]	80.42	58.57	338.56	30.21
	LIN	8.12	4.66	2.83	0.03
	QUAD	4.76	2.56	6.11	2.96
	Proposed	6.89	4.01	2.91	0.05

Table 3.6 Performance evaluation in terms of bit rate using test sequences with QP=30

Error Models		Test sequences (rate, %)			
		<i>Foreman</i>	<i>Carphone</i>	<i>Mobile</i>	<i>Container</i>
QP=30	FULL [kbps]	35.78	23.52	79.80	10.89
	LIN	0.17	1.02	2.19	0.07
	QUAD	0.14	0.04	2.73	0.09
	Proposed	-0.36	-0.19	2.43	0.09

Switching the error criterion models

The performance of the proposed switching scheme has been evaluated in terms of bit rate by averaging over first 100 frames in the H.263 framework, using test video sequences including “Foreman”, “Carphone”, “Mobile”, “Container”, as shown in Figure 3 and Table 3.5-3.6. In the tables, LIN and QUAD correspond to error criterion models, linear and quadratic respectively, and FULL means a two stages search, where a three step search is adopted in integer pixel level and 8 surrounding pixels are searched for the best vector.

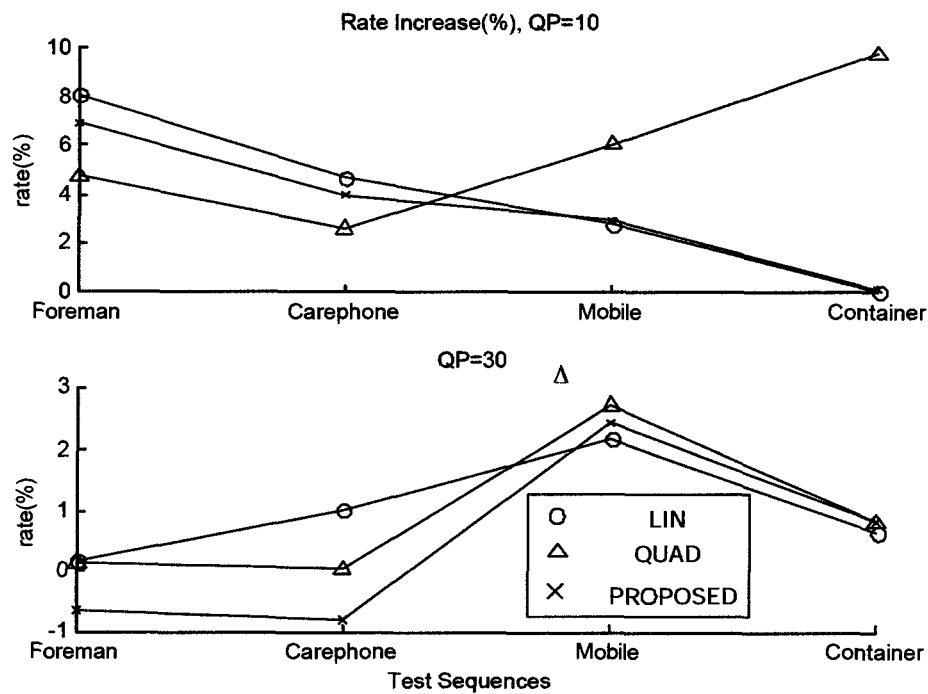


Figure 3.7 Performance in relative increase of bit rate compared to the full search (%)

Let d_m be the difference between the estimated values from models and the actual computed values in the integer pixel search, and $m \in \{1 = LIN, 2 = QUAD\}$ represents each model. The difference d_m is shown in Figure 3.6 and can be represented as

$$d_m = abs(d_{m-}) + abs(d_{m+}) \\ = |\varepsilon(-2) - \bar{\varepsilon}(-2)| + |\varepsilon(2) - \bar{\varepsilon}(2)|$$

Then, the process of model switching is described as

$$m = \arg \min_{m \in \{1 = LIN, 2 = QUAD\}} (d_m)$$

A model with minimum difference is chosen as the best among two models for the motion vector search in the current location.

In the experimental data shown in Table 3.5 and Table 3.6, neither of LIN and QUAD models does not necessarily performs the best with all video sequences. In other words, LIN work better than QUAD for “*Fore-man*” and “*Car-phone*”, while QUAD works better with “*mobile*” and “*sales-man*” when the quantization parameter is set $QP=10$. It is also noteworthy that the model performance gets affected by the quantization parameter in the coding framework. Coding performance has been changed with “*Container*” input according to the quantization parameter QP .

Experiments have verified that the proposed scheme improves the video coding performance in terms of bit rates up to 3% compared to other method with a given sequence, which is the case for “*Mobile*” with $QP=10$ as shown in Figure 3.7. Consequently, the proposed method is proven to be more efficient and accurate in the performance of motion estimation process.

3.5 SUMMARY

The error criterion function of motion estimation is well represented by a mathematical expression such as the quadratic and linear models around the optimal point. The error criterion function leads to the sub-pixel accuracy motion estimation in two stages process (for example, in case of half-pixel accuracy, full pixel search and interpolation at half sub-pixel accuracy). In the paper, explicit solutions are derived based on a linear model function [68].

The pre-computed error criterion values being computed at full-pixel level are used to derive the motion vector and the error criterion values at the sub-pixel accuracy. Hence, the approach reduces dramatically the number of computations compared to conventional methods, where the error criterion function at the sub-pixel accuracy is computed directly from interpolated sub-pixel values.

The maximum gradient between two error criterion values at the integer pixels leads to a direction geometrically, where the optimal point is closely located among the horizontal, vertical, and two diagonal directions. The proposed gradient method is shown to further improve the motion estimation performance, while the complexity increase is negligible. In addition, a novel approach switching to one of two models according to a metric has been introduced in the paper. The method was proven that it leads to better performance up to 3% compared to other methods with test sequences. It needs to be verified in rate control framework as a future work.

Chapter 4

REGRESSIVE MODEL BASED RATE DISTORTION OPTIMIZATION

Both motion estimation and residual quantization coding are jointly optimized using a rate-distortion model so that the overall computation complexity can be significantly reduced, though it incurs a small sacrifice in rate-distortion performance. Generally a rate-distortion optimization requires excessively complex operations associated with motion vector decisions, DCT, and quantization operations. We formalize its problem, and then propose a simplified approach for a practical implementation purpose. It gives a substantial reduction of computational complexity by changing the joint optimization problem associated with the motion vector and the quantization parameter to two sequentially dependent optimization problems. The proposed scheme is a fast and efficient implementation of a rate-distortion optimized motion estimation algorithm, where model parameters are estimated by a linear regression algorithm and updated dynamically within the predefined frame window and according to varying input video sequences. For complexity comparison, it is estimated in terms of the number of required RISC-like instructions, and then compared to those of the RD-optimal and the conventional MSE optimal algorithm. Experimental results show that the proposed adaptive model approach closely approximates the optimal performance, while significantly reducing the required computational complexity. Furthermore, the proposed method outperforms the conventional MSE optimal method in terms of PSNR performance and computational complexity.

4.1 INTRODUCTION

In conventional video coding systems, motion vectors are selected by considering only distortion and then the quantization parameter is optimized for the vector so as to meet either a bit-rate or a distortion constraint. In other words, it takes into account the residual error only in motion vector decisions excluding the residual error bit-rate generated after the quantization of the residual error. Such a motion vector estimation is easily affected by noise sources such as camera noise and illumination change, which incur a large number of bit allocations for motion vector representation. Furthermore, motion vector bit-rate takes a substantial portion of the overall bit rate in low bit-rate video coding applications. The overall bit-rate needs to be optimally allocated between the motion vector coder and the residual coder, in order to avoid performance degradation resulting from ill-conditioned bit-rate allocation. From this perspective, motion vector estimations based on rate-distortion measures can lead to the overall improvement of system performance, since the saved bit-rates in motion estimations can be efficiently spent on coding the residual error.

The optimal rate-distortion optimization algorithm requires excessive computations because it performs DCT and scalar quantization operations for each candidate motion vector and quantization parameter. To achieve a computational complexity reduction, a set of parametric rate and distortion functions is introduced to estimate the rate and distortion values in a small sacrifice of performance. As a matter of fact, the optimization algorithm based on rate distortion function, which is well suited to the properties of video sequences, can achieve near optimal performance. Generally, operational rate distortion functions are obtained through the preprocessing of the test video sequence in offline applications. However, it is not realistic to preprocess the video sequence to evaluate its rate distortion function in the real-time implementation due to constraints on the allowable delay.

In the past, lots of research was carried out in order to reduce the computational complexity of the rate distortion optimized algorithm. Interpolation techniques [55] and table look up-methods [56, 57, 60, 61] were implemented to reduce the complexity in estimating rate and distortion information. In interpolation techniques, the number of rate distortion evaluations that are expensive computationally are reduced by limiting calculations to only predefined points. Then, evaluations for inter sample points are driven by the interpolation. Look-up table approaches were made under the assumption that rate and distortion performance is uniquely determined by quantization parameters and residual error. In other words, these methods commonly evaluate rate distortion functions through offline preprocessing of the test video sequence and keep using fixed functions through the entire sequence. Furthermore there was no adaptive scheme applied in the past approaches. In the chapter, we propose a fast and efficient rate-distortion optimization with an adaptive model, where model parameters are estimated by using a linear regression algorithm and applied to input video sequences real-time. The proposed algorithm reduces the excessive computation complexity associated with motion vector decisions, DCT, and quantization operations. In particular, it updates model parameters dynamically within a predefined frame window and according to varying input video sequences.

This chapter is organized as follows. In section 4.2, a rate distortion problem is formulated and its implementation is discussed from a real-time application point of view. In section 4.3, an adaptive rate-distortion optimization algorithm is proposed, where rate and distortion function is derived and modeled by the 2nd order least mean square (LMS) method. An adaptive control of model parameters is explained in the H.263 framework, which keeps accurately tracking the varying properties of input video sequences. In section 4.4, the computational complexity of the proposed model-based method is analyzed and compared to RD optimal and TMN5. Experimental results based on the proposed algorithm are presented in section 4.5, and conclusions follow in section 4.6.

4.2 PROBLEM FORMULATION

In conventional motion-compensated video coding systems, motion vectors are estimated by searching the minimum of the matching criteria, such as mean absolute error (MAE) or mean squared error (MSE). Consider a frame consisting of N macroblocks. Let $\vec{d} = (d_1, \dots, d_N)$ and $q = (q_1, \dots, q_N)$ represent the motion vector set and the quantization parameter set, respectively. Then, MAE can be described as follows,

$$\min_{\vec{d} \in S} \sum_{\vec{r} \in W} |I(\vec{r}, n) - I(\vec{r} + \vec{d}, n-1)| \quad (4.1)$$

where $I(\vec{r}, n)$ is the intensity of each pixel of frame n and \vec{r} is its coordinate, and S and W are, respectively, the search area and the macro block of 16×16 .

The conventional cost measure does not take into account the overall rate and distortion in the motion estimation stage, which can potentially lead to a loss of system performance. Therefore, we consider the motion vector coder and the residual error coder jointly, and use a general definition of cost measure $D_{overall}(\vec{d}, q)$, which is the distortion after the residual coding. It is expressed as a function of the motion vector and quantization parameter. Here a problem is how to find the motion vector and quantization parameter which minimizes the overall distortion for a given bit-rate constraint. This can be formulated as shown below.

$$\min_{\vec{d}, q} \sum_{k=1}^N D_{overall}^k(\vec{d}, q) \quad (4.2)$$

$$\text{subject to } \sum_{k=1}^N R_d^k(\vec{d}) + \sum_{k=1}^N R_e^k(q) \leq R_{Total}$$

where N and R_{Total} are respectively the total number of macro blocks and the given bit-rate constraint for the current frame. This hard-constrained problem can be solved efficiently by converting it into an unconstrained problem by the Lagrangian optimization method, where rate constraint is merged with overall distortion through the Lagrange multiplier λ . Then, the converted unconstrained problem can be written below.

$$J_{total}(\lambda) = \min_{\vec{d}, q} \sum_{k=1}^N J_k(\vec{d}, q) \quad (4.3)$$

$$\text{where } J_k(\vec{d}, q) = D_{overall}^k(\vec{d}, q) + \lambda \left(R_e^k(\vec{d}, q) + R_d^k(\vec{d}) \right)$$

The optimal R-D point minimizing the total Lagrangian cost function can be searched through the convex hull of the operational R-D, which is estimated by preprocessing the input video sequence. The Lagrange multiplier λ , which controls overall rate and distortion, is set to the negative slope of the line tangent to the obtained R-D curve at the operating point.

In fact, searching the solution of the optimization problem is an intensively demanding computational operation, since it involves joint optimization between motion estimation and residual coding. In other words, the DCT and quantization operation should be performed for each motion vector over the search windows to evaluate the rate and distortion term of (4.3). Such computational requirements are infeasible in most practical implementations.

In sequential optimization, motion estimation is associated with the residual coding stage. Since the terms $D_{overall}^k(\vec{d}, \vec{q}) + \lambda_d R_e^k(\vec{d}, \vec{q})$ of (4.4) are proportional to the residual error in the motion estimation $D_{mv}^k(\vec{d})$, we can ignore the effect of residual error coding in the motion vector estimation by using an approximation $D_{overall}^k(\vec{d}, \vec{q}) + \lambda_d R_e^k(\vec{d}, \vec{q}) \propto D_{mv}^k(\vec{d})$. In order to simplify the computation overhead further, all macroblocks are treated independently, although MV's are coded differentially. As a result, ignoring the existing dependency incurs a small loss of performance, but leads to a sub-optimal solution. Furthermore, the quantization parameter q_i is assumed fixed in the macroblock layer. Taking into account the approximation with this assumption, the constrained equation can be rewritten as follows [62].

$$\sum_{k=1}^N \min_{\vec{d}} \left\{ D_{mv}^k(\vec{d}) + \lambda_d R_d^k(\vec{d}) \right\} \quad (4.5)$$

$$\min_{q_i} \sum_{k=1}^N \left\{ D_{overall}^k(\vec{d}, q_i) + \lambda_q \left(R_e^k(\vec{d}, q_i) + R_d^k(\vec{d}) \right) \right\}$$

Although the computation complexity of the rate and distortion optimized algorithm was reduced substantially through a sequence of simplification procedures, (4.5) is still computationally intensive for practical application, since the DCT and quantization operation should be calculated for the given optimal motion vector. In order to alleviate this prohibitive computational requirement, a parametric model based approach is introduced to estimate rate distortion performance.

4.3 RATE DISTORTION FUNCTION MODELING

The direct estimation of rate and distortion requires DCT of the residual error, and the quantization associated with all combinations of motion vectors and quantization parameters. This intensive computational requirement makes the implementation of the rate-distortion optimization algorithm impractical. Assume that motion vectors are given as the simplified approach derived in the previous section. The residual bit-rate and the distortion can be estimated from models approximated by simple second order polynomial functions. Hence, a model-based method is adopted here to circumvent such a massive computational operation. Generally, a stochastic model consists of two rate and distortion prediction functions with respect to the quantization parameter and provides, respectively, estimations of rate and distortion that result from the encoding of the residual error. It is well known that, with respect to the increasing quantization parameter, the rate function of the statistical model decreases monotonically while the distortion function increases monotonically.

Let $\{X_1, \dots, X_N\}$ be the DCT transform coefficients of residual error in a 8x8 block with $N = 64$. To illustrate the parametric modeling approach to rate and distortion, we assume distortions of the squared quantization error have a Gaussian distribution. Applying the theoretical rate distortion function, the block level rate and distortion functions of the quantization parameter $q \in \{1, \dots, 31\}$, $R(q)$ and $D(q)$ are derived as follows [63, 64].

$$R(q) = \sum_{i=1}^N \frac{1}{2} \log_2 \frac{\sigma_i^2}{\alpha_i q^2} \propto \frac{1}{q^2} \quad (4.6)$$

$$D(q) = q^2 \sum_{i=1}^N \alpha_i \propto q^2 \quad (4.7)$$

where σ_i^2 is the variance of X_i and α_i is constants. These model functions are approximated by the parametric functions of the quantization parameter q . Note that the block-level model can be extended to the frame-level by accumulating the rate and distortion for the total number of macroblocks without affecting the fundamental formula. Here the frame-level rate and distortion model functions, $R(q)$ and $D(q)$ can be approximated as follows.

$$R(q) = a_1 q^{-1} + a_2 q^{-2} \quad (4.8)$$

$$D(q) = b_1 q^1 + b_2 q^2 \quad (4.9)$$

where q is the quantization parameter, and a_1 , a_2 , b_1 , and b_2 are model coefficients. In fact, the properties of video sequences vary in time. As a result, the fixed model parameters can not represent properly the rate distortion performance of the variant input sequence. Hence, it is necessary to update the model parameters adaptively in order to reflect the changing input characteristics. An adaptive model-based approach is implemented here so that the number of bits corresponding to residual error, motion vector, and syntax information, as well as the distortion for the current frame, can be more optimally predicted using the observed values from the most recent frames.

In general, the properties of video sequences vary slowly in low bit-rate applications, such as video conferencing. Hence, the estimation of the model parameters can be computed and updated by the least mean square (LMS) method [58, 59] on the basis of recent observations. The model coefficients a_1 , a_2 , b_1 , and b_2 of rate and distortion model functions, $R_{i+1}(q)$ and $D_{i+1}(q)$ in the frame $i+1$, can be calculated from the actual encoding results in the past frames $\{i-n-1, \dots, i\}$ within the predefined frame

window n using a linear regressive analysis. The calculations of model coefficients are described as follows [58].

$$a_2 = \frac{n \sum_{i=1}^n r_i - (\sum_{i=1}^n q_i^{-1})(\sum_{i=1}^n q_i r_i)}{n \sum_{i=1}^n q_i^{-2} - (\sum_{i=1}^n q_i^{-1})^2} \quad (4.10)$$

$$a_1 = \frac{\sum_{i=1}^n q_i r_i - a_2 q_i^{-1}}{n}$$

$$b_2 = \frac{n \sum_{i=1}^n d_i - (\sum_{i=1}^n q_i^{-1})(\sum_{i=1}^n q_i d_i)}{n \sum_{i=1}^n q_i^{-2} - (\sum_{i=1}^n q_i^{-1})^2} \quad (4.11)$$

$$b_1 = \frac{\sum_{i=1}^n q_i d_i - a_2 q_i^{-1}}{n}$$

where q_i , r_i , and d_i are the quantization parameter, the bit rate, and the distortion from the actual encoding in the past frames respectively. The encoder collects and keeps the bit rate, distortion and the quantization parameter of frames within the predefined sliding-frame window defined by n . Model parameters are updated after encoding each frame by applying LMS adjustment on a data set consisting of the most recent observations. These procedures are described in detail as follows:

Step 1: Initialize model parameters a_0 , a_1 , b_0 , and b_1 based on data collected from frames in the beginning.

Step 2: Encode a frame with frame number i and collect parameters q_i , r_i , and d_i for quantization, rate, and distortion, respectively.

Step 3: Calculate model parameters in (4.10) and (4.11), and update the model in (4.8) and (4.9) to be used for the next frame.

Step 4: Increase the frame number $i = i + 1$ and go back to the step 2. Repeat the step 2 to step 4 until the end of sequence.

Note that the size of sliding-frame window n represents the number of frames to be considered in the parameter calculation, and it can be adjusted, based on the required adaptability and activity of video sequences. Furthermore, the resulting estimated model function can be checked for its monotonicity in the range of possible quantization parameters before being adopted as the new control parameters. This verification process leads to more a reliable estimation of model parameters by withdrawing invalid parameters resulting from the abrupt variation of video properties.

4.4 COMPLEXITY ISSUES

The computational complexity of the proposed model-based approach is compared with that of the RD optimal and the TMN5. Assume that motion vector search range is $[-15, 15]$ and video sequences are in QCIF(176x144 pixels) formats. In the model-based and the optimal RD methods, motion vector and quantization parameters are searched through a joint rate-distortion procedure, based on the introduced Lagrangian cost function, while the TMN5 uses conventional distortion-based criteria such as MAE. To be fair in the following comparisons, we assume that the motion vector search is conducted exhaustively for all possible candidate vectors, although there are many fast approximate methods available for not only conventional MAE-based methods but also for the RD

optimization algorithms. In the MAE distortion-based approach, fast algorithms such as TSS[10], 2-D LOG[9], DS[11, 12], and Conjugate Directional Search(CDS)[42] are commonly used. The so-called MV-pruning methods [57, 60] can be applied in RD optimization algorithms since the RD optimal motion vector is usually located near the motion vectors found by MAE distortion-based methods, as well as the motion vectors of the surrounding macroblocks.

In the previous formulation of the RD optimization method, we assumed that the joint optimization problem between the motion vector and the quantization parameter could be simplified by two sequentially dependent optimization problems as shown in (4.5), where the optimization of motion estimation and residual coding can be conducted independently. Subsequently, the required number of computational operations can also be estimated simply by the use of independent complexity analyses in motion estimation and residual coding.

In regard to the computational complexity of motion estimation which corresponds to the first part of (4.5), the Lagrangian cost for a possible candidate motion vector requires MAE distortion and MV rate calculations. The MAE cost function requires 2×256 load operation, 256 subtraction operation, 256 addition operation, 1 division operation, 1 store operation, and 1 data compare operation, for a total 1035 operations. The MV rate can be obtained by a table look-up and three arithmetic operations, for a total of 4 operations [12]. Consequently, the computational operation of rate and distortion calculations for a vector C_{lag} is described below.

$$C_{lag} = C_{mae} + C_{mvr} = 1035 + 4 = 1039 \quad (4.12)$$

where C_{mae} and C_{mvr} represent the number of operations for MAE distortion, MV rate calculation respectively. Assuming that the search range is given by $[-p, p]$, the total

number of search points is equal to $(2p+1)^2$. For instance, when the search range p be 15, the total number of search points is 961. In the same manner, the computational complexity of the motion estimation in (4.5) a frame C_{me} is calculated as follows.

$$C_{me} = C_{lag} \times N_{sp} \times N_{mb} = 1039 \times 961 \times 99 = 98,849,421 \quad (4.13)$$

where C_{lag} , N_{sp} , and N_{mb} represent the number of operations for calculation of the Lagrangian cost for a vector, the number of search points, and the number of macroblocks respectively. On the other hand, in the residual error coding which corresponds to the second part of (4.5), the calculation of the Lagrangian cost for a possible candidate motion vector requires the following operations: a DCT of the residual error, the rate calculation through the quantization operation of the DCT coefficients, zig-zag scanning, VLC, and the MSE distortion calculation through IDCT. For the sake of simplified complexity estimation, we take into account only major coding modules, including DCT, IDCT, and MSE distortion calculations. Assume that a row-column decomposition method among many DCT algorithms is used on 8x8 blocks. Its computational operation requires eight data loads, eight DCT coefficients, eight multiply-accumulate operations, and one data store operation for a total of 25 operations a pixel data [12, 65]. The number of operations becomes $2 \times 25 \times 64 = 3,200$ for a 8x8 2-D block. Therefore, the total operations result in $4 \times 3200 = 12,800$ for a macroblock of 16x16 pixels. On the other hand, the MSE cost function requires a total of 1035 operations that include 2x256 loads, 256 subtractions, 256 multiply-accumulates, 1 division, 1 store and 1 data comparison. By taking into account the results, the computational operations for DCT/IDCT and MSE are 12,800 and 1035, respectively.

Since the quantization parameter $q \in \{1, \dots, 31\}$ is used in H.263 video coder, the MSE distortion calculation is repeated 31 times, respectively, for each quantization parameter

with a given motion vector. Hence, the required computation of a given motion vector C_{res_mv} is written as follows.

$$C_{res_mv} = C_{DCT} + 31 \times (C_{IDCT} + C_{mse}) = 12,800 + 31 \times (12,800 + 1,035) = 441,685 \quad (4.14)$$

where C_{DCT} , C_{IDCT} , and C_{mse} represent the number of operations for DCT, IDCT and MSE respectively. The computational complexity of the residual error coding a frame C_{res} is $441,685 \times 99 = 43,726,815$. Therefore, the total complexity a frame C_{rd} is given by

$$C_{rd} = C_{res} + C_{me} = 43,726,815 + 98,849,421 = 142,576,236 \quad (4.15)$$

which is equal to both the complexities of motion estimation and of residual error coding in (4.5). Taking into account the total number of frames, either 10 or 30 frames/sec, it shows that the required computational operations are too intensive, especially for real implementation.

These massive computational operations of rate and distortion in residual error coding can be alleviated by the rate and distortion model functions introduced in the previous section. As shown in (4.8) and (4.9), the rate and distortion estimation for a motion vector requires only 4 multiplications and 2 arithmetic operation to calculate the Lagrangian cost. Therefore, the complexity of (4.14) reduces to $31 \times 6 = 186$, and the computational complexity a frame is $186 \times 99 = 18,414$. Since motion estimation is conducted without any change, the total number of computational operations C_{rd_model} results in $18,414 + 98,849,421 = 98,867,835$ a frame. On the other hand, the model-based approach needs to conduct a sequence of model parameter update frame by frame, requiring about 200 operations in (4.10) and (4.11), with averaging window size $n = 10$.

Table 4.1 Computational complexity for the model-based, RD optimal and TMN5 with the motion vector search range (-15, 15)

Algorithms	Number of computational operations/frame	Computational Ratio
RD-optimal	142,576,236	1.00
Model based	98,868,035	0.69
TMN5	98,468,865	0.69

Consequently, the total number of computational operations per frame in the model-based approach C_{rd_model} is given by

$$C_{model} = C_{total} + 200 = 98,867,835 + 200 = 98,868,035 \quad (4.16)$$

In the TMN5, MAE distortion is defined as a cost function. Taking into account the same number of operations for the MAE calculation as given above, the total complexity a frame C_{tmn5} is 74,113,281 for the exhaustive motion vector search. Its calculation can be described by

$$C_{tmn5} = C_{mse} \times N_{sp} \times N_{mb} = 1035 \times 961 \times 99 = 98,468,865 \quad (4.17)$$

where C_{mse} , C_{ns} , and C_{mb} represent the number of operations for cost function, the number of search locations per macroblock, and the number of macro blocks per frame respectively. The total computational operations are compared in Table 4.1 for the model-based approach, the RD optimal, and the conventional distortion respectively.

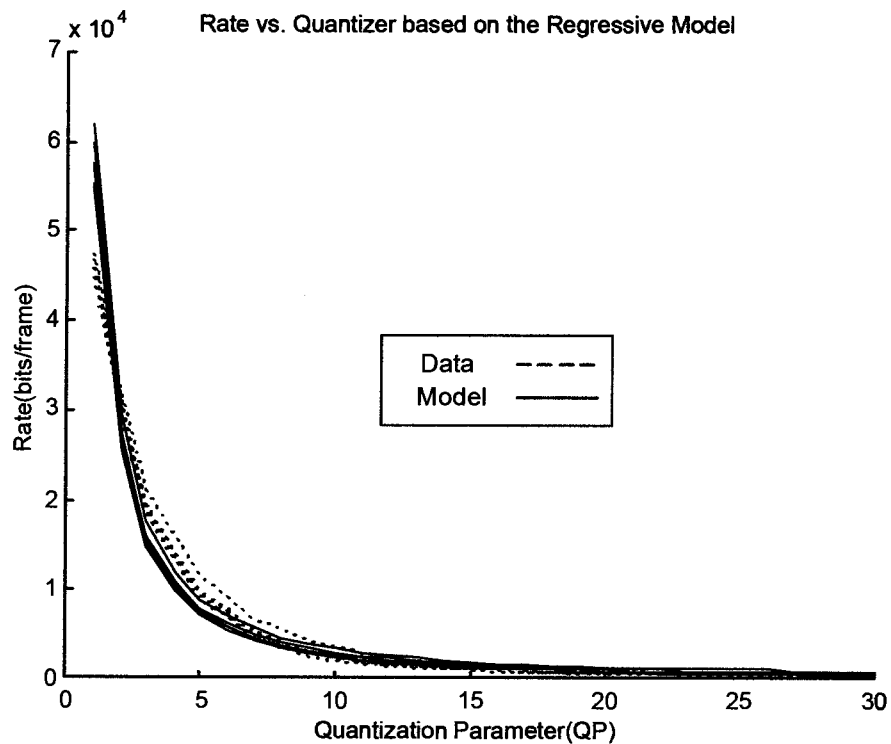


Figure 4.2 Rate function approximated by the 2nd order regressive model for the first five frames in the sequence *Miss- America*

The complexity of the proposed model-based approach requires about 31% fewer computations than that of the RD-optimal approach, while being comparable to that of the conventional distortion-based method. Note that in the table the complexity of the RD-optimal method is not significantly different from that of the conventional distortion-based method, due to the simplified implementation of the RD optimization algorithm. Therefore, the computational ratio would be different from the result shown in the table when the complexity is estimated for the original RD optimizations.

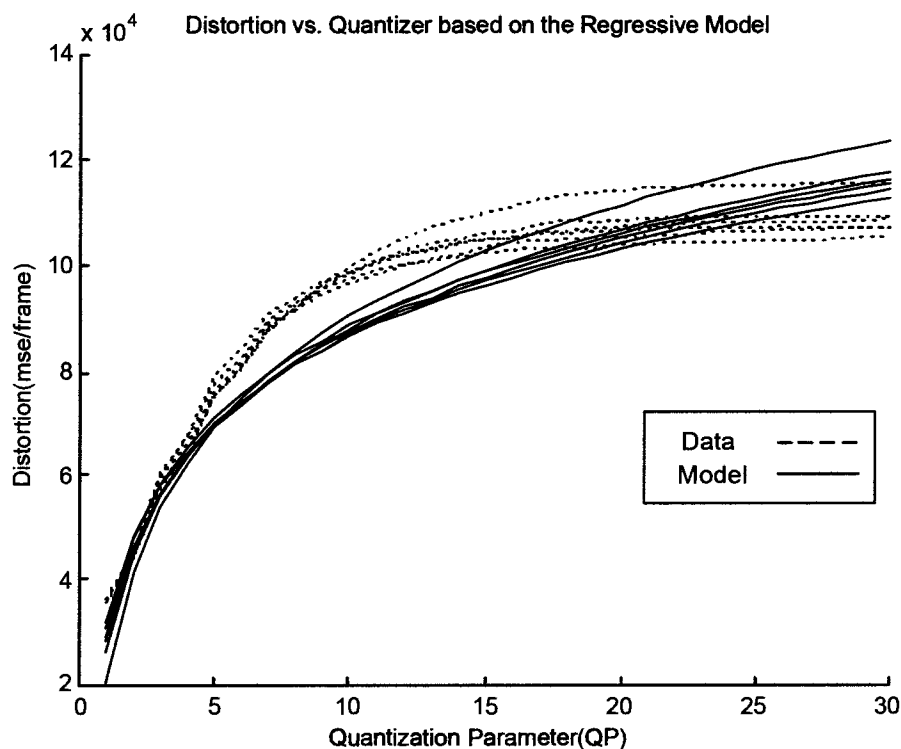


Figure 4.3 Distortion function approximated by the 2nd order regressive model for the first five frames in the sequence *Miss-America*

4.5 EXPERIMENTAL RESULTS

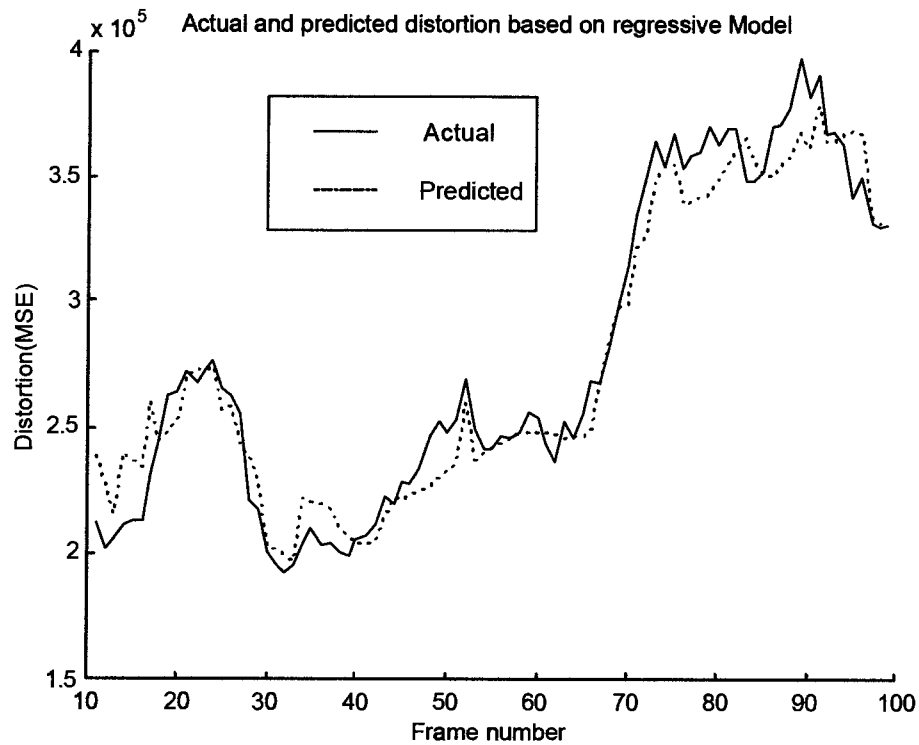
The fast and efficient rate distortion optimization algorithm was introduced in the previous section, where rate and distortion values for the quantization parameter were estimated from the approximated rate and distortion functions, rather than actual calculations involving computationally expensive DCT and quantization operations, among others. In particular, rate and distortion functions are adaptively changed through the updated control parameter by the linear regressive method (LMS), which reflects the varying properties of input video sequences.

Table 4.2 Relative rate and distortion model error in RMSE using different averaging window size of the regression model, with the video sequence *Miss-America*

Average window(n)	Rate model	Distortion model
2	2.1253e+3	9.2972e+4
3	2.1042e+3	8.8603e+4
4	2.0807e+3	9.5946e+4
5	1.9650e+3	1.0240e+5
10	2.1022e+3	1.0948e+5
20	2.1279e+3	1.4567e+5

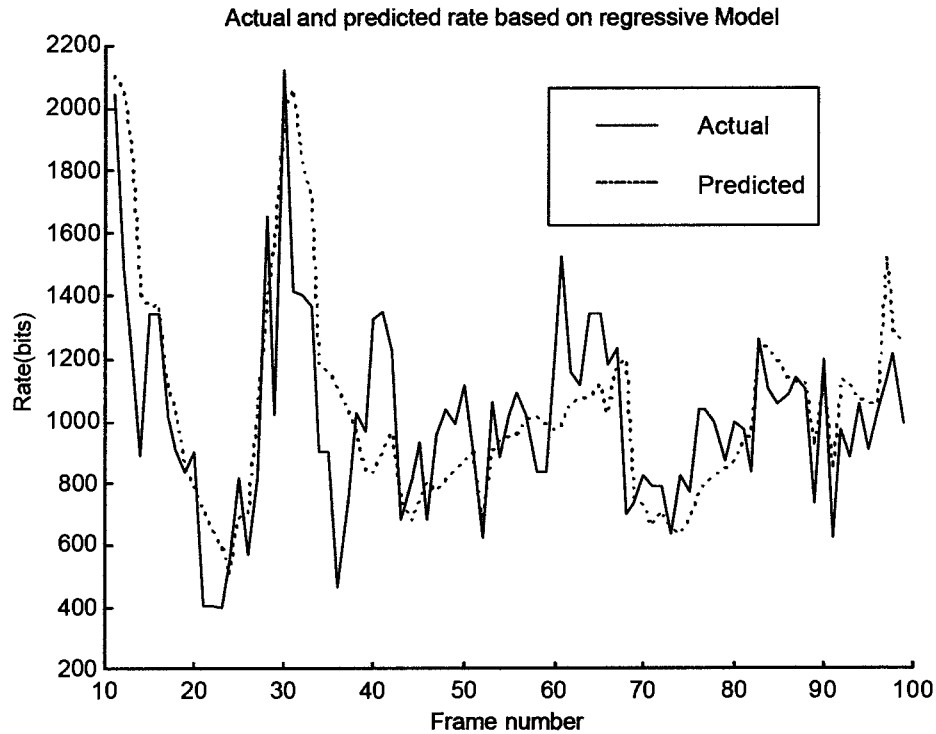
This proposed rate-distortion method is compared to both conventional and optimal RD optimization algorithms in terms of its computational complexity and performance. Note that the rate distortion function is modeled with respect to only the quantization parameter in the frame layer in order to evaluate the proposed adaptive model-based algorithm, although it can be extended to macro block layer with the increased complexity involving the search of both optimal motion vector and quantization parameter.

The following experiments were conducted using *Miss-America* and *Carphone* sequences in the H.263 video coding framework. The sequences have moderate motion in scene activity, and meet the low bit-rate condition assumed in the rate distortion model. Let the picture size be QCIF (176x144) and its frame rate be adjusted to 10fps by skipping two frames in the original video sequence at 30fps.



(a) Distortion

Rate and distortion performance was measured by averaging the first 50 frames in the following experiments. First, the proposed rate and distortion function approximated by 2nd order parametric function is evaluated in its accuracy. As shown in Figure 4.2 and 4.3, the rate and distortion model converges closely with real rate and distortion data, which were obtained from the first 5 P-frames of *Miss-America*, in the H.263 coding framework. To compare the performance of the adopted model according to a different averaging frame window size, relative rate and distortion model error are estimated in RMSE defined as shown below



(b) Rate

Figure 4.4 Actual and predicted distortion (a) and rate (b), based on regressive model with the averaging window size 10, and with the video sequence *Miss-America*

$$RMSE = \sqrt{\sum_{i=1}^N (y_i - \tilde{y}_i)^2} \quad (4.16)$$

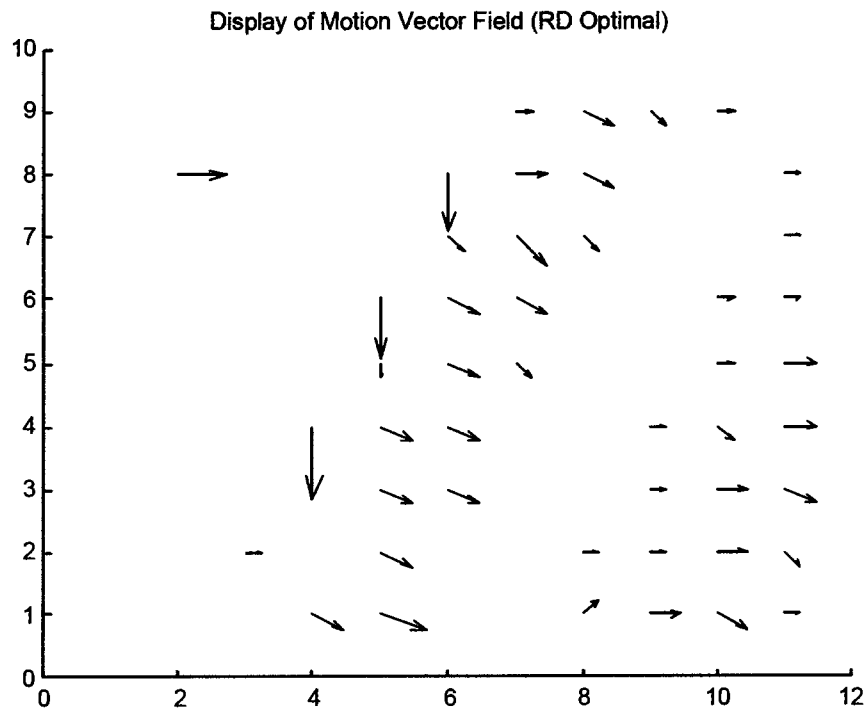
where y_i and \tilde{y}_i are the actual and predicted values, respectively. Experimental results using video sequence *Miss-America* are shown in Table 4.2, where the rate and distortion model gives the best result in RMSE with the averaging window size 5 and 3, respectively.

Table 4.3 Rate constrained motion estimation in terms of average rate [bits/frame] and PSNR, QP=15, frames = 50, 10fps

Lagrangian Multiplier λ_d	<i>Miss-America</i>			<i>Carphone</i>		
	Overall Rate	MV rate	PSNR	Overall Rate	MV rate	PSNR
0	766	198(0.25)	35.61	1907	426(0.22)	31.36
10	723	151(0.20)	35.46	1836	357(0.19)	31.33
20	714	138(0.19)	35.36	1819	317(0.17)	31.30
50	741	118(0.15)	35.14	1843	263(0.14)	31.17
100	862	105(0.12)	34.80	1946	214(0.10)	31.06

For an example, Figure 4.4 graphically represents the regressive model that tracks down the actual rate and distortion data with a time delay, where the averaging frame window size is set to 10 frames. It becomes evident in the results that the averaging window size of the LMS model also needs to be adaptively changed in the video sequence in order to improve the overall system performance.

As shown in the dependent optimization problem of (4.5), the rate-distortion constrained motion estimation can be further simplified to reduce its computational complexity. In fact, this optimization problem is complicated by the dependency, since motion vectors are differentially coded from the predicted one using the surrounding vectors.



(a) RD optimal

Generally, its optimal solution can be found through the Dynamic Programming approach. For the sake of simplicity and uncomplicated implementation, the dependency of motion vectors is ignored in the rate-constrained motion estimation. In other words, over the search range, each motion vector is searched independence from the surrounding vectors over the search range, which leads to a locally optimal motion vector for the current macroblock. Table 4.3 shows the motion vector rate and the distortion under rate-distortion constraints defined by the Lagrangian multiplier λ_d , with the video sequences *Miss-America* and *Carphone*.

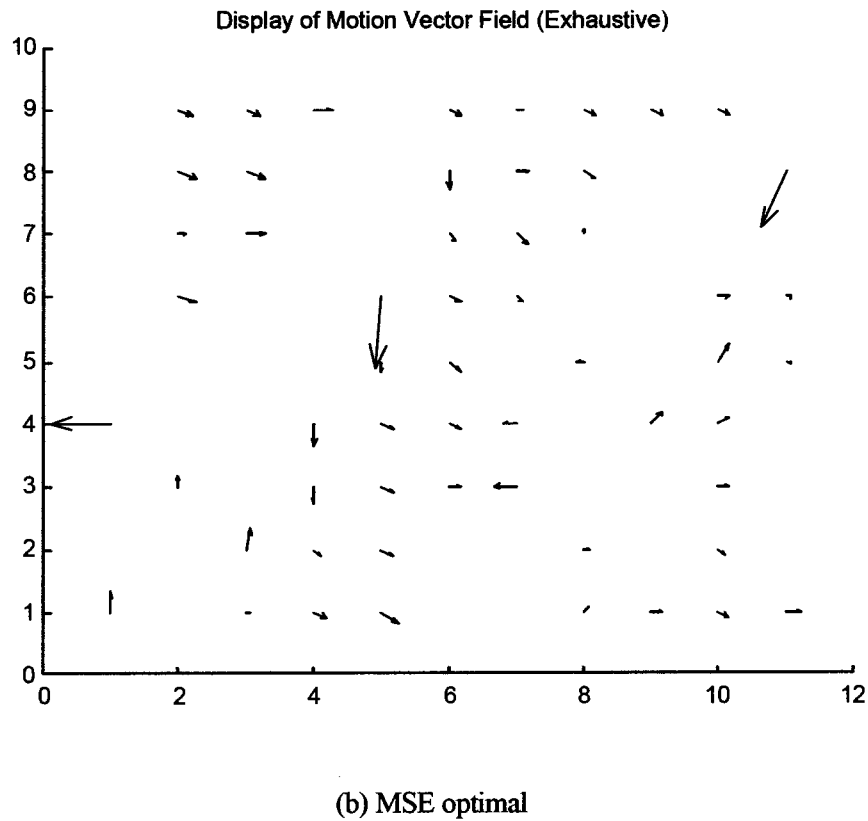
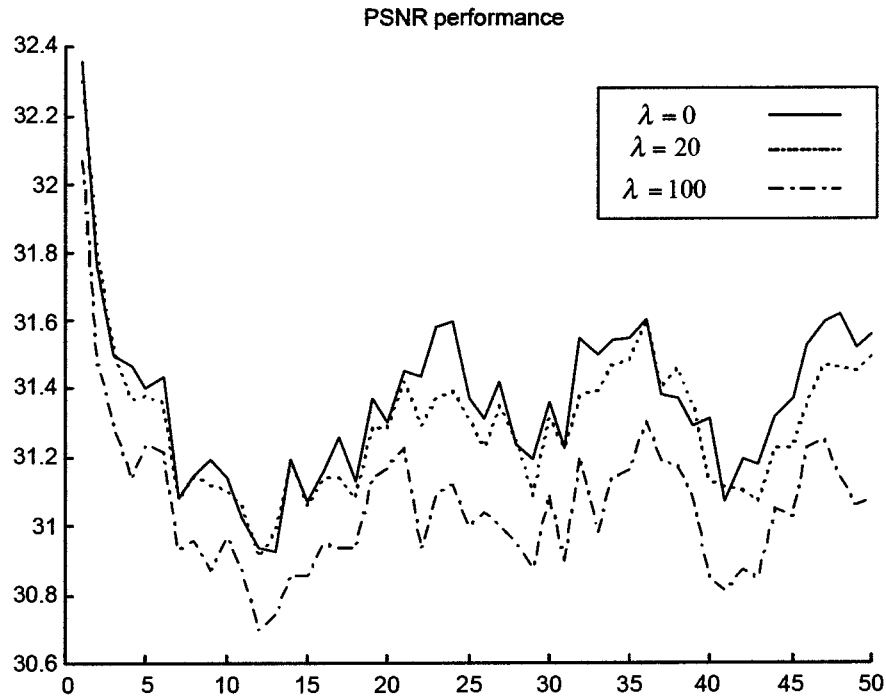


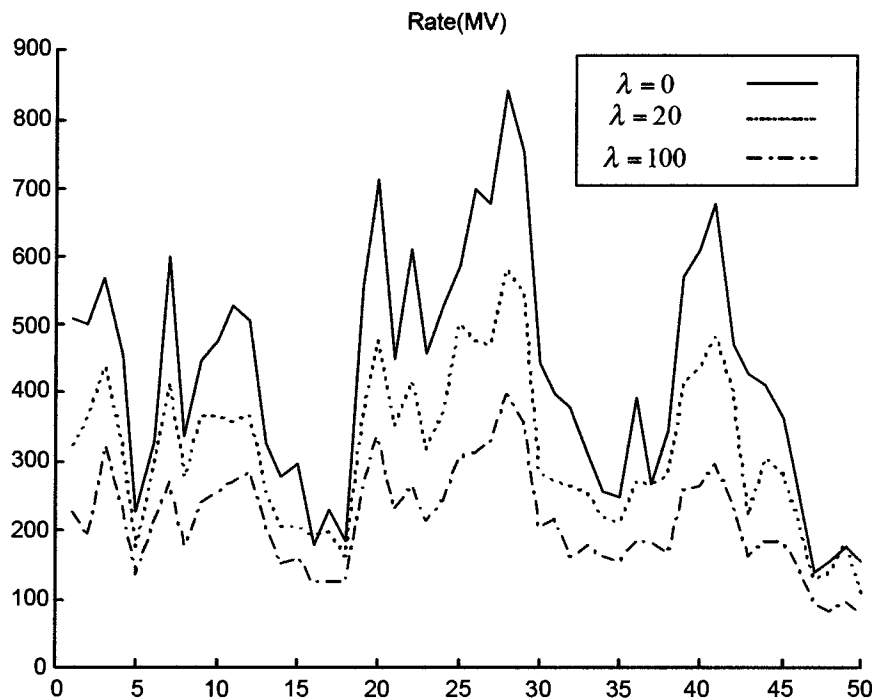
Figure 4.5 Comparison of motion vector field between rate-constrained (a) and exhaustive full search (b) motion estimation methods with the frame number =10, QP=15, the video sequence *Carphone*

As shown in the Table 4.3, bit-rate reduction can be achieved with a small sacrifice of PSNR performance. Note that the best performances are achieved with $\lambda_d = 20$ for both sequences. Therefore, the Lagrangian multiplier λ_d in the rate-distortion optimal motion estimation is assumed to be 20, unless otherwise given with a specific value in the following experiments.



(a) PSNR

Motion vector fields were compared in Figure 4.5. The motion vector fields of the rate-distortion constrained method become smoother than those of the MSE optimal motion estimation method. The smoother motion field reduces the required bit-rate, since motion vectors are differentially coded from the predicted one, based on surrounding motion vectors. Moreover, it is shown that the rate distortion constrained method removes the noisy motion field, which is often found in the background area of the scene by the MSE optimal method. As an instance of this, Figure 4.6 shows PSNR and MV bit-rate changes for the video sequence *Carphone*, with the quantization parameter $QP = 15$ and 50 total frames.



(b) MV rates

Figure 4.6 PSNR performance and MV bit-rates according to the given rate constraints 0 to 100, with $QP = 15$, 50 total frames, and the video sequence *Carphone*

The effects of rate-distortion optimal motion estimations were investigated in terms of MV smoothness and PSNR. Now we estimate the overall performance of the proposed algorithm wherein a sequence of optimization occurs for motion estimation and quantization parameter selection. For the rate distortion optimal motion estimation, the Lagrangian multiplier $\lambda_d = 20$ is assumed in the following experiments. First, the relation between the averaging window size n of the regressive model and the overall distortion performance was investigated in terms of PSNR.

Table 4.4 Performance comparisons in terms of PSNR according to the different averaging window size using *Miss-America* and *Carphone* sequences

Averaging Window Size, n	Video Sequences	
	Miss-America (4 kbps), PSNR [dB]	Carphone (10 kbps), PSNR [dB]
1	36.22	29.97
2	36.22	29.99
3	36.22	29.99
4	36.22	30.02
5	36.22	30.05
6	36.21	30.00
7	36.19	29.99
8	36.21	29.99
9	36.21	30.01
10	36.21	29.99
Adaptive	36.22	29.99

Note that the averaging window size is fixed through the sequence. It did not affect the overall performance significantly in experiments with two video sequences, *Miss-America* and *Carphone*, as shown in Table 4.4. Only small differences less than 0.1dB were observed when the averaging size n increases 1 to 10.

Table 4.5 Rate distortion performance using the sequence *Miss-America*

Algorithms	4kbps	6kbps	8kbps	10kbps
Optimal RD	36.34	37.58	38.24	38.79
Model-based	36.22	37.28	37.92	38.46
TMN5	35.95	36.98	37.81	38.33

Let σ_i^2 represent the input variance of residual error in the frame i . Then, the variance σ_i^2 is defined by

$$\sigma_i^2 = \frac{1}{(N \times N)} \sum_{i=1}^{N \times N} (d_i - \tilde{d}_i)^2 \quad (4.18)$$

where d_i and \tilde{d}_i represent the pixel intensity of residual error at a location i and its average in a macroblock with the size $N \times N$ respectively. When the averaging size n was changed between 1 and 2 frame by frame according to the input variance of residual error σ_i^2 , its output performance consistently is shown better than those obtained with the size n fixed throughout the video sequence. We assumed that the average variance for each video sequence $\sigma_{threshold}^2$ was available from off-line processing since it can have a different value, depending on the input video sequences. The update equation used in determining the averaging window size is described as follows:

Table 4.6 Rate distortion performance using the sequence *Carphone*

Algorithms	10kbps	20kbps	30kbps	40kbps
Optimal RD	30.14	32.31	33.64	34.71
Model-based	29.99	32.18	33.55	34.60
TMN5	29.84	32.03	33.33	34.41

Step 1: Initialize parameters $i = 1$ and $n_i = 1$ for frame number and averaging window size, respectively.

Step 2: Encode a frame i and compute the input residual variance σ_i^2

Step 3: Compare σ_i^2 with $\sigma_{threshold}^2$, and update the averaging window size n_{i+1} by

$$\begin{aligned}
 n_{i+1} &= 1 \text{ if } (\sigma_i^2 > \sigma_{threshold}^2) \\
 &= 2 \text{ otherwise}
 \end{aligned}
 \tag{4.19}$$

Step 4: Increase $i = i + 1$ and go back to the step 2. Repeat the step 2 to the step 4 until the end of sequence.

where $\sigma_{threshold}^2$ is the average variance of each input video sequence obtained by off-line processing.

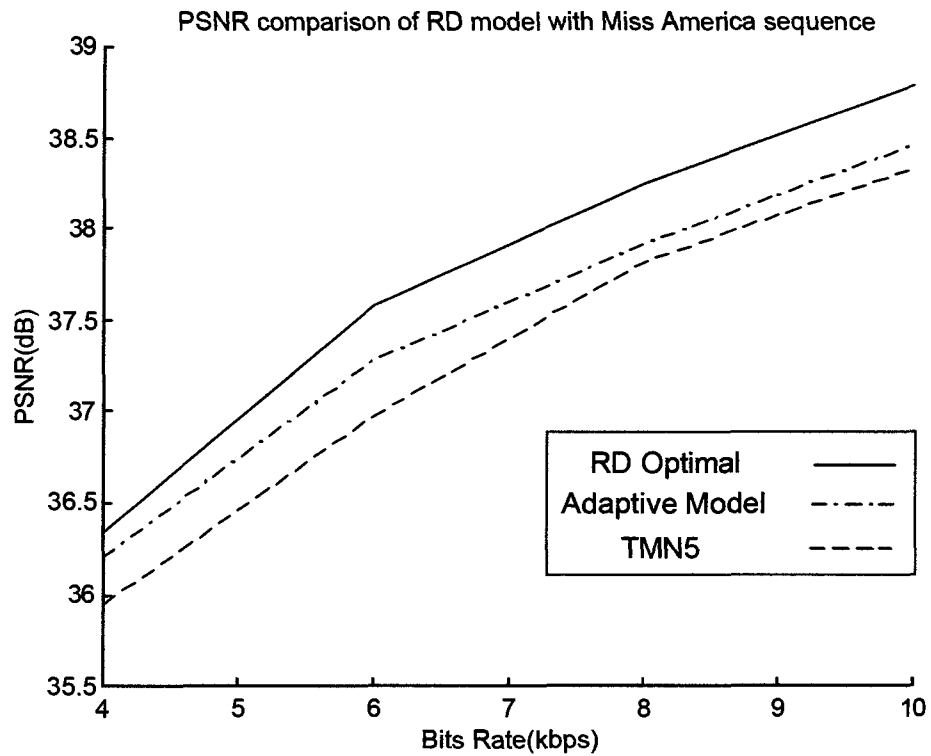


Figure 4.7 PSNR performance of rate distortion model with *Miss-America* sequence

We compared the overall rate distortion performance of the proposed model-based approach with both TMN5 and optimal RD optimization methods. Note that the TMN5 is implemented based on the analytical model, while the optimal RD optimization method searches for optimal rate distortion conditions through the exhaustive method. For the model-based approach, the Lagrangian multiplier $\lambda_d = 20$ and the averaging window size $n = 2$ are assumed in following experiments.

The experimental procedure is described as follows:

Step 1: Initialize RD model parameters with initial parameters $\lambda_d = 20$, $i = 1$, $q_i = 12$, and $n_i = 2$ for the Lagrangian multiplier, quantization parameter, frame number, and averaging window size, respectively.

Step 2: Compute RD optimization equation (4.5) using the RD model in (4.8) and (4.9).

Step 3: Compute R_i and D_i for a frame i and update the RD model parameters.

Step 4: Increase $i = i + 1$, and go back to the step 2. Repeat the step 2 to step 4 until the end of sequence.

Figure 4.7 and Figure 4.8 show the performance of rate distortion optimization expressed in terms of PSNR when the average consumed bits range from 4kbps to 10 kbps, and from 10kbps to 40 kbps, respectively. The corresponding data are shown in Table 4.5 and Table 4.6.

It is proven that the performance of the proposed adaptive model-based algorithm is close to that of the optimal R-D algorithm, and better than that of TMN5. Experimental results show that the optimal RD optimization algorithm has the best performance among the three algorithms, with average differences of about 0.6dB from TMN5, and 0.2dB from the proposed adaptive method in terms of PSNR, although the optimal rate distortion algorithm is much too complex to be implemented in real video coding applications. It is noteworthy that the proposed method keeps tracking that of the optimal rate distortion algorithm with the same bit usage, while its computational complexity is relatively negligible in comparison to that of the optimal RD optimization.

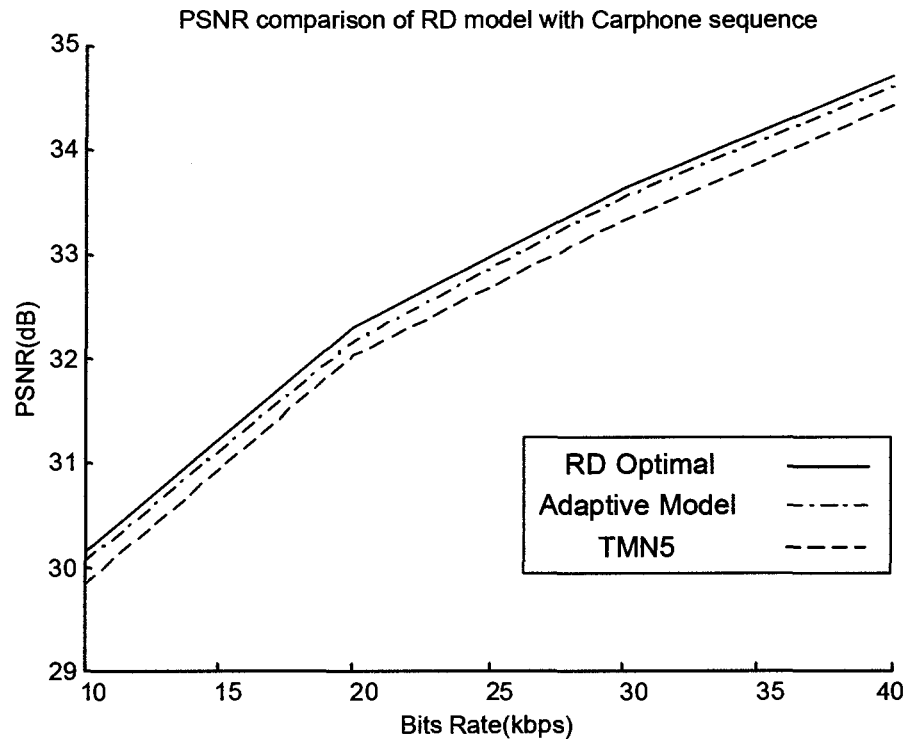


Figure 4.8 PSNR performance of rate distortion model with *Carphone* sequence

4.6 SUMMARY

It was shown that overall rate and distortion performance could be improved close to that of the optimal algorithm. Through a fast rate distortion optimization algorithm, quantization parameter and motion vector are optimally chosen so as to minimize residual bit-rates and distortion. The parametric approximation model of rate and distortion function consisting of quantization parameter is used for the estimation of real rate and distortion value. This results in a substantial reduction of computational complexity relevant to DCT and quantization, while it incurs a small sacrifice in rate-distortion performance. On the other hand, an adaptive scheme is introduced in the model and its control parameters are updated in accordance to varying input sequences. For the sake of

performance evaluation, the optimization problem was simplified as two sequential dependent problems so that the motion vector and the quantization parameter could be searched independently for their optimal values. Note that in the simplified approach, the rate and distortion function was evaluated with respect to the quantization parameter of frame layer. However, this experiment would be extended to the macroblock layer optimization by considering motion vector, quantization parameter, and coding mode in future research.

Chapter 5

DISTORTION AND COMPLEXITY OPTIMIZATION IN SCALEABLE VIDEO CODING SYSTEM

A configurable coding scheme is proposed and analyzed with respect to computational complexity and distortion. The major coding modules are analyzed in terms of computational complexity and distortion (C-D) in the H.263 video coding framework. Based on the analyzed data, operational C-D curves are obtained through an exhaustive search, and the Lagrangian multiplier method. The proposed scheme satisfies the given computational constraint independently of the changing properties of the input video sequence. A technique to adaptively control the optimal encoding mode is also proposed. The performance of the proposed technique is compared with a fixed scheme where parameters are determined by off-line processing. Experimental results demonstrate that the adaptive approach leads to computation reductions of up to 19%, which are obtained with test video sequences and compared to the fixed, while the PSNR degradations of the reconstructed video are less than 0.05dB.

5.1 INTRODUCTION

Multimedia communications involving audio, video and data has been an interesting topic because of the many possible applications. Recently, hardware platforms for hand-held devices such as PDAs have improved dramatically, which has created a special interest in implementing videos in portable devices. However, video-coding algorithms are still much too complex for implementation in hand-held devices, which are powered by batteries with a limited storage capacity. Therefore, computationally configurable video coding schemes would be beneficial for such constrained environments.

The question is how to achieve optimal computing resource allocation among encoding modules for given computational constraints, so that the system can make the best use of limited computing resources to maximize its coding performance in terms of its video quality. Work in the area of optimal video coding is reviewed in [1, 2]. One of the common approaches is to optimize the bit allocation by taking into account the resulting rate and distortion. Although this is a good approach to deal with bandwidth limitations, this may not give good performance where the computational complexity is the main limitation.

The rate distortion optimization problem in a video coding framework is addressed in [3, 4], where motion estimation, mode decision, and quantization are considered either separately or jointly for the best tradeoff. Although complexity is addressed in conjunction with rate and distortion, only the DCT and IDCT modules of the video coding system are considered [5, 6].

In this paper, the performance of a configurable video system is analyzed with respect to computational complexity and distortion. The system consists of three coding modules, each having a control parameter (such as window size in Motion Estimation) controlling the computational complexity and the quality of the reconstructed video sequence. The approach considered here is different from the one in [7], where an iterative method is used to find the optimal control variables. More specifically the method in [7] measures the system complexity in terms of averaged fps, while the one proposed in [5, 6] gives the predetermined complexity of the coding system regardless of the varying input contents and sequence. [65] introduces a baseline framework of the proposed concept and presents interim results. Based on the previous work, we here extend it to an adaptive scheme whereby more accurate control parameters are found particularly with active sequences.

This approach could be reasonably accurate enough to estimate the system complexity as far as major coding modules are taken into account in the system configuration. The complexity and distortion data is obtained by analyzing the operations required for each module, and by evaluating the distortion in the reconstructed sequence for the possible control parameter values.

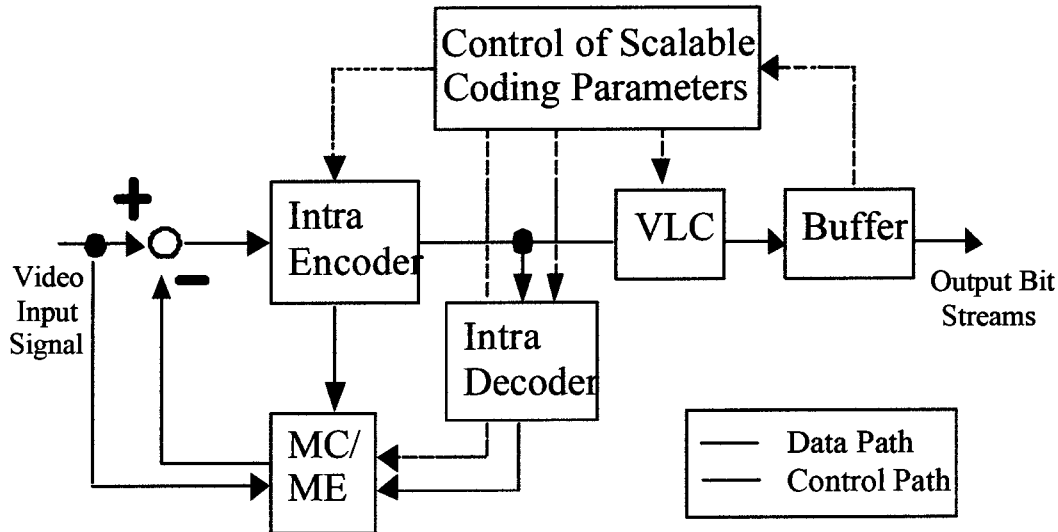


Figure 5.1 Configurable coding scheme with scalable coding parameters

This paper is organized as follows. In section 5.2, a general formulation of the optimization problem is presented. In section 5.3, the computational complexity and distortion of major coding modules are analyzed. An operational Complexity-Distortion (C-D) curve is obtained using the analyzed data from test video sequences, and an adaptive control scheme is introduced in section 5.4. Finally, its implications for the performance of the coder are discussed, and concluding remarks given, in section 5.5.

5.2 GENERAL PROBLEM FORMULATION

Consider a video coding system that is decomposed in N modules M_1, \dots, M_N . Each module M_i , $i = 1, \dots, N$, is assigned a control variable s_i , which determines both the computational complexity required for coding and the distortion of the reconstructed video sequence. Each control variable s_i can take k_i distinct values from the set $S_i = \{s_{ij} \mid j = 1, \dots, k_i\}$ for $i = 1, \dots, N$. With these definitions, it is now possible to express the computational complexity $C(s_1, \dots, s_N)$ for the video coding system as

$$C(s_1, \dots, s_N) = \sum_{i=1}^N c_i(s_i) \quad (5.1)$$

where $c_i(s_i)$ is the computational complexity for each coding module M_i , $i = 1, \dots, N$. The complexity for each coding module depends on the control variable for this module s_i .

The distortion between the original and the reconstructed video sequence can be represented as $D(s_1, \dots, s_N)$. Each coding module M_i , $i = 1, \dots, N$, contributes to $D(s_1, \dots, s_N)$ even though the individual contributions are not additive. The distortion depends again on the control variable s_i for each module M_i .

The problem considered here is finding the control variable values for the N coding modules, which would lead to minimal distortion of the reconstructed video sequence for a given limited computational complexity. This can be formulated as follows:

$$\min_{[s_1, \dots, s_N] \in \mathcal{S}_1 \times \dots \times \mathcal{S}_N} D(s_1, \dots, s_N) \quad (5.2)$$

subject to $C(s_1, \dots, s_n) \leq C_{\max}$.

This is a constrained optimization problem where the optimization variable s_1, \dots, s_N can take distinct values. A known approach[26, 27, 28, 29, 30, 31] to solve this constrained optimization problem is to consider the following unconstrained optimization problem.

$$\min_{[s_1, \dots, s_N] \in \mathcal{S}_1 \times \dots \times \mathcal{S}_N} D(s_1, \dots, s_N) + \lambda C(s_1, \dots, s_N) \quad (5.3)$$

where the Lagrangian multiplier λ is a nonnegative number. It is well known in operational research that the Lagrangian relaxation method will not necessarily give the optimal solution, since Lagrangian multiplier λ can reach only the operating points belonging to the convex hull in the operational complexity-distortion curve. When λ sweeps from 0 to infinity, the solution to problem (5.3) traces out the convex hull of the complexity distortion curve. The Lagrangian multiplier λ allows a trade-off between complexity and distortion performance. When $\lambda = 0$, minimizing the Lagrangian cost function is equivalent to minimizing the distortion. Conversely, when λ changes to infinity meaning becomes large enough, and minimizing the Lagrangian cost function is equivalent to minimizing the complexity. Many fast algorithms have been developed by many authors[32, 33, 34] to find the optimal λ . Hence, assuming an optimal Lagrangian multiplier for the given computational constraint is given through either a fast or an exhaustive search of the Lagrangian multiplier, the problem now is to find the optimal solution to the unconstrained problem of (5.3).

In this thesis, a configurable video coding scheme like the one outlined in Figure 5.1 is considered. For our analysis it is assumed that the system consists of three major coding modules with corresponding control variables:

M_1 : Motion Estimation(ME) module where the control variable s_1 can take values from the set $S_1 = \{0, \dots, 3\}$ corresponding to variable search range, $p \in \{3, 5, 7, 9\}$, respectively.

M_2 : Integer or Fractional(I/F) pixel accuracy in ME, where the control variable can take the values $s_2 = 0$ (integer) or $s_2 = 1$ (fractional) pixel accuracy

M_3 : DCT where the control variable s_3 can take values from the set $S_3 = \{0, \dots, 3\}$ corresponding to different DCT coefficient pruning options $w \in \{2, 4, 6, 8\}$, respectively.

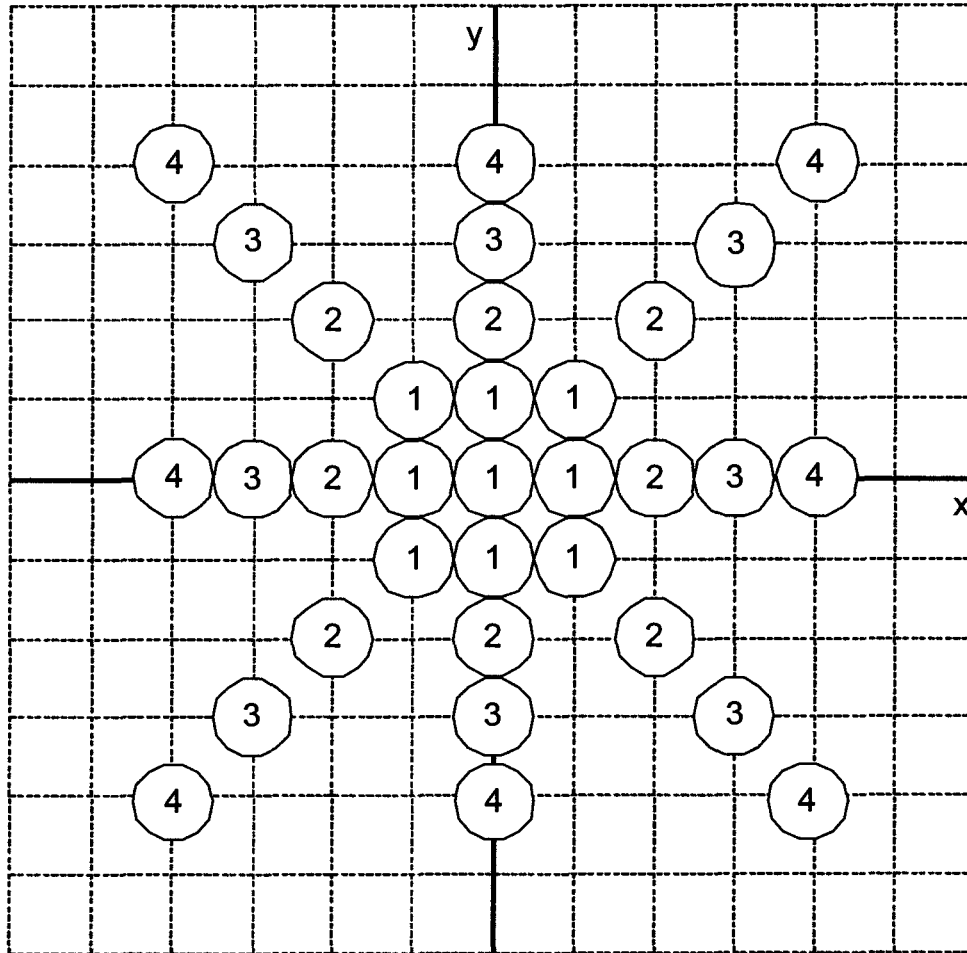


Figure 5.2 Search points according to the different search window in the Three-Step Search

5.3 COMPLEXITY AND DISTORTION ANALYSIS

In this section, the computational complexity of each of these coding modules is evaluated. Among the various metrics possible, the approach, which considers all instructions, including multiplications and additions with the same weighting as one instruction, will be used here[12]. Since we are interested in the relative complexity and accuracy, the computational complexity for only one frame is computed.

ME module

There are many block-matching fast search algorithms, such as TSS[10], 2-D LOG[9], DS[11, 12], Conjugate Directional Search(CDS)[42], and so on, which have been developed to reduce the computational complexity of a full exhaustive search algorithm. TSS is one of the fast search algorithms, reducing computational complexity to $8 \log p$, where p is the search range parameter. The size of the initial step, and the next, is calculated by dividing the search range parameter p by 2 in each. The number of search points is eight in each step, except in the initial one, which needs one more point in the zero vector location.

Note that the computational complexity of TSS given in the number of search points is constant, not changing with the varying contents in the video sequence. In TSS, the search points are pre-defined for all macroblocks, as shown in the figure. Other algorithms, such as DS and CDS, search for the motion vector of the macroblock starting from the zero vector location until the best motion vector is found that meets the given cost measure, the locations and the total number of search points change for each macroblock.

This deterministic property can be used in implementing a configurable coding system with a hard-control feature. Therefore, this search range parameter is chosen as a control parameter in a tradeoff between complexity and accuracy. Figure 5.2 shows the number of search points with regard to the search range, where zero vector $MV(0,0)$ is assumed as the real vector giving the minimum cost function. The numbers 1, 2, 3, and 4 in the figure, which mean the window size of the motion vector search, correspond to 3×3 , 5×5 , 7×7 , and 9×9 , respectively.

Table 5.1 Computational complexity as a function of the search window size for the ME search used

Search Windows size, s_1	$s_1=0$	$s_1=1$	$s_1=2$	$s_1=3$
	[-3,3]	[-5,5]	[-7,7]	[-9,9]
Search Points	9	17	25	33
Computations	694,089	1,311,057	1,928,025	2,544,993

The complexity analysis here is based on a frame size of 176×144 QCIF format, a block size of 16×16 and the use of the Mean Absolute Difference (MAD) as the matching criterion. The MAD calculation can be represented as below.

$$MAD(dx, dy) = \frac{1}{N^2} \sum_{i=-\frac{N}{2}}^{\frac{N}{2}} \sum_{j=-\frac{N}{2}}^{\frac{N}{2}} |F(i, j) - G(i + dx, j + dy)| \quad (5.4)$$

where $F(i, j)$ is the $N \times N$ macroblock being compressed; $G(i, j)$ is the reference $N \times N$ macroblock, and dx and dy are the search location motion vectors; N is the macroblock size. The evaluation of each MAD cost function requires 2×256 load operations, 256 subtraction operations, one division operation, one store operation and one data compare operation, for a total $2 \times 256 + 256 + 1 + 1 + 1 = 1035$ operations [12].

The overall computational complexities according to different search ranges are analyzed in Table 5.1.

I/F module

The accuracy of the motion vectors obtained can be improved using half pixel accuracy[10]; that is, by using 8 surrounding half-pixels from the integer pixel location. First, computing operations for bilinear interpolation per macro block are 324 data loads, 162 additions, 162 divisions, 486 data accumulations and 162 data divisions, for a total of 1296 operations. Therefore, for the QCIF format and block size of 16x16, the total number of operations for a half-pel search can be evaluated as follows.

$$\begin{aligned} & (\text{Total number of operations per MAD cost function} \times \text{Number of search locations} \\ & \text{surrounding integer motion vector} + \text{Bi-linear interpolation per integer motion vector}) \times \\ & (\text{Number of macro blocks}) = (779 \times 8 + 1296) \times 99 = 745,272 \end{aligned} \quad (5.5)$$

DCT module

DCT has been used for most image and video coding standards because its energy compaction performance is close to that of Karhunen-Loeve Transform (KLT), known as the optimum image transform in terms of energy compaction, sequence entropy and de-correlation. Most of the energy is compacted into the top left corner, so that the least number of elements are required for its representation. The basic computation of the DCT-based video and image compression system is the transformation of an 8x8 image block from the spatial domain to the DCT transform domain. The 2-D 8x8 transformation is expressed as [14]

$$y(k,l) = \frac{c(k)c(l)}{4} \sum_{i=0}^7 \sum_{j=0}^7 x(i,j) \cos \frac{(2i+1)k\pi}{16} \cos \frac{(2j+1)l\pi}{16}, k,l = 0,\dots,7 \quad (5.6)$$

where $c(k) = \frac{1}{\sqrt{2}}$ for $k=0$ and $c(k) = 1$ otherwise.

The 2-D DCT transform can be decomposed into two 1-D 8-point transforms, as (5.6) can be modified as

$$y(k,l) = \frac{c(k)}{2} \sum_{i=0}^7 \left[\frac{c(k)}{2} \sum_{j=0}^7 x(i,j) \cos \frac{(2j+1)k\pi}{16} \right] \cos \frac{(2i+1)l\pi}{16}, k, l = 0, \dots, 7 \quad (5.7)$$

where $[\cdot]$ denotes the 1-D DCT of the rows of input $x(i, j)$.

Regarding computational complexity, the 2-D DCT computation of the equation (5.6) requires 4096 multiplications and additions. However, using the row-column decomposition approach of (5.7), it can be reduced to 1024 multiplications and additions, four times less than that of (5.6). Although the separability property of DCT has reduced the computational complexity, these numbers are still prohibitive for real-time application. Until now, many fast DCT computation algorithms [20, 21, 22] have been developed utilizing transform matrix factorization as well as previously developed Fast discrete Fourier Transform (FFT). However, since the quantizer follows the DCT computation unit in most image and video coding systems, its computational complexity can be further reduced. All of the multiplication occurring in the last stage of transform can be absorbed into the following quantizer unit. In other words, this computation yields the scaled version of real DCT output. The computational complexities of the most commonly used fast DCT algorithms can be analyzed in the scaled-DCT approach [22].

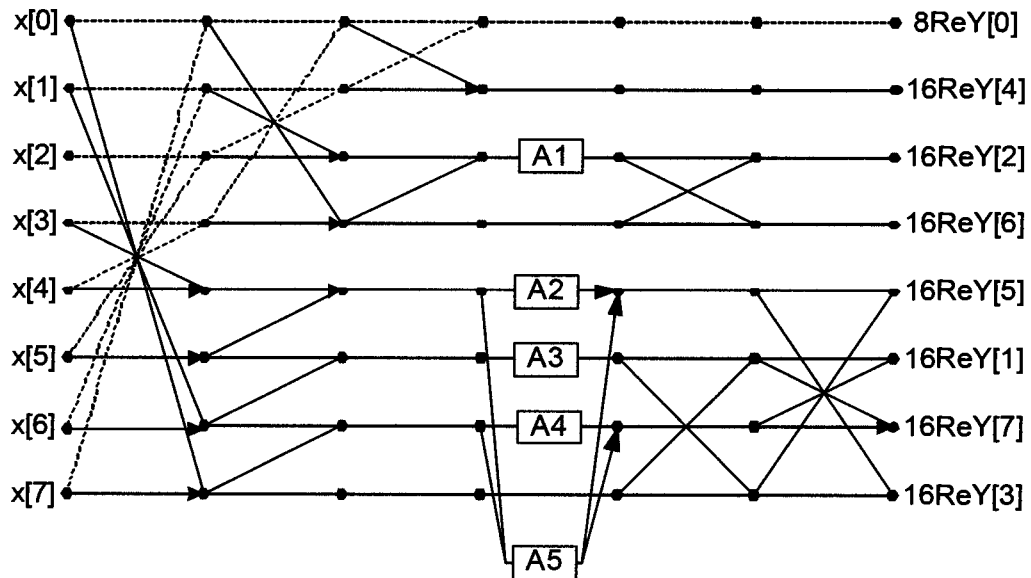


Figure 5.3 AAN forward DCT flow chart where DCT pruning for $y(0)$ coefficient is represented by the dotted line

AAN scheme [33], adopted for the implementation of DCT pruning in this section, is the fastest implementation among the scaled 1-D DCT algorithms. It adopts the small and fast FFT algorithm developed by Winograd requiring only 5 multiplications and 29 additions, and is expressed as

$$y(k) = \frac{2c(k)\text{Re} Y(k)}{\cos \frac{n\pi}{16}} \quad (5.8)$$

where $c(k) = \frac{1}{\sqrt{2}}$ for $k=0$ and $c(k) = 1$ otherwise, and $\text{Re} Y(k)$ are the real part of the 16-point DFT, whose inputs are double sized, with inputs $x(k), k = 0, \dots, 7$.

Table 5.2 Computation complexity as a function of pruning for the DCT module

Complexity, s_3		2x2 Pruning			4x4 Pruning			6x6 Pruning			Full DCT		
		$s_3=0$			$s_3=1$			$s_3=2$			$s_3=3$		
		M	A	T	M	A	T	M	A	T	M	A	T
AAN	ID	3	18	21	5	23	28	5	27	32	5	29	34
	8x8	400			588			742			880		
	Frame	158400(0.45)			232848(0.67)			293832(0.84)			348480(1.00)		

Its flow chart for forward DCT calculation is shown in Figure 5.3. Note that for real DCT data, outputs of the flow graph should be multiplied by constants in the equation (5.8). However, these multiplications, can be absorbed into the quantization process, giving overall computation reduction since DCT outputs are quantized for compression in most video and image coding systems.

One property of the DCT transform is efficient energy compaction, and the Human Visual System (HVS) is no more sensitive to high frequency components than the low frequency ones. These facts can be used to make computation-intensive DCT transform scaleable and controllable in its computational complexity. Some of the DCT coefficients can be pruned, since they do not need to be calculated at all. The DCT pruning reduces the computational complexity of the DCT transform, since it has an efficient energy compaction property and the most important information is kept in the low frequency coefficient. The dotted line in Figure 5.3 shows required computations when DCT pruning is applied to the $y(0)$ transform coefficient, where a total of seven additions are needed. Pruning DCT transform is studied in [23, 24].



(a) 2x2(25.660dB)



(b) 4x4(30.650dB)



(c) 6x6(31.739dB)



(d) 8x8 full DCT(31.740dB)

Figure 5.4 Reconstructed video frames with DCT coefficient pruning (QP=13, Intra I-frame, and H.263)

A transform [23] derives an analytical form of computational complexity, where DCT pruning is applied to a fast 1-D DCT algorithm [25] with 12 multiplications and 29 additions. However, in this paper, AAN DCT is adopted in the computational complexity analysis of DCT pruning, since it is the best among the known 1-D DCT algorithms.

In [14], algorithmic complexity of the 2-D DCT algorithm is analyzed using row-column decompositions, which performs 1-D DCT two times for each of the rows and columns of 8x8 input data. A similar complexity measure can be applied to the AAN algorithm [22]. Table 5.2 shows the number of operations required to compute the DCT coefficients for each 8x8 block, and a frame of QCIF format when different pruning is used. In the Table, 1D and 8x8 mean 1D 8-point and 2D 8x8 DCT, respectively. It estimates the number of multiplications and additions as well as the total sums, with the assumption that the same weighting factor is given to both multiplication and addition.

In Figure 5.3, 1D 8-point DCT requires eight data loads, five DCT coefficients, eight data stores, five multiplications, and twenty-nine additions, for a total of 55 operations. Therefore, in the 8x8 2-D block, the total number of operations becomes $2 \times 8 \times 55 = 880$ operations. It also shows how much DCT pruning performs the relative reduction of computation compared to the 8x8 full DCT.

The DCT pruning basically discards high frequency components in the transform domain, although it incurs image quality degradation. Figure 5.4 shows reconstructed video frames after the DCT pruning operation. More coefficients are pruned, and more quality degradation occurs in the reconstructed frames. It is interesting to note that applying DCT pruning with a 4x4 window or an 8x8 full DCT makes little difference in terms of subjective quality, although there is a difference in the objective performance of about 1.1dB PSNR. This can be explained by the fact that the DCT has a property of high efficient energy compaction, and most energy is concentrated in the upper left corner. Accordingly, the computational complexity of DCT can be traded off with the reconstructed image quality using, the DCT pruning.

Table 5.3 Average PSNR data and computational complexity of all operation modes, where five video sequences were applied and their results were averaged

Operation Mode			Average PSNR (dB) $D(s_1, \dots, s_N)$	Overall Computations ($1.0e+6$, %) $C(s_1, \dots, s_N)$
ME s_1	I/H s_2	DCT s_3		
0	0	0	28.13	0.734(21.72)
		1	30.53	0.752(22.27)
		2	31.21	0.768(22.73)
		3	31.48	0.781(23.13)
	1	0	29.06	1.479(43.79)
		1	31.56	1.498(44.34)
1	0	0	28.21	1.351(39.99)
		1	30.55	1.369(40.54)
		2	31.26	1.385(40.99)
		3	31.54	1.398(41.40)
	1	0	29.11	2.096(62.06)
		1	31.59	2.115(62.61)
2	0	0	28.22	1.968(58.26)
		1	30.56	1.986(58.81)
		2	31.29	2.001(59.26)
		3	31.55	2.015(59.67)
	1	0	29.12	2.713(80.33)
		1	31.59	2.732(80.88)
3	0	0	28.22	2.585(76.53)
		1	30.54	2.603(77.08)
		2	31.29	2.618(77.53)
		3	31.55	2.632(77.93)
	1	0	29.12	3.330(98.59)
		1	31.60	3.348(99.14)
3	0	0	28.22	2.585(76.53)
		1	30.54	2.603(77.08)
		2	31.29	2.618(77.53)
		3	31.55	2.632(77.93)
	1	0	29.12	3.330(98.59)
		1	31.60	3.348(99.14)
3	0	0	28.22	2.585(76.53)
		1	30.54	2.603(77.08)
		2	31.29	2.618(77.53)
		3	31.55	2.632(77.93)
	1	0	29.12	3.330(98.59)
		1	31.60	3.348(99.14)

The overall computational complexity $C(s_1, \dots, s_N)$ can be calculated from the equation (5.1) and the above discussion, while the overall distortion $D(s_1, \dots, s_N)$ can be estimated by exhaustive simulation for all possible operation modes of control variables, and averaged over a number of sequences and a number of frames for each sequence. In the given system, there are total 32 modes consisting of combinations of the three control variables s_1 , s_2 , and s_3 , corresponding to ME, I/H and DCT, respectively. Table 5.3 shows the overall computation and distortion data for all 32 operating modes. Computational complexities are represented in a total number of RISC-like instructions per frame, while distortions are measured in the peak signal-to-noise ratio (PSNR) as follows:

$$PSNR = 10 \log_{10}(255^2 / MSD) \quad (5.9)$$

$$MSD = \frac{1}{N} \sum_{i=1}^N (O_i - R_i)^2 \quad (5.10)$$

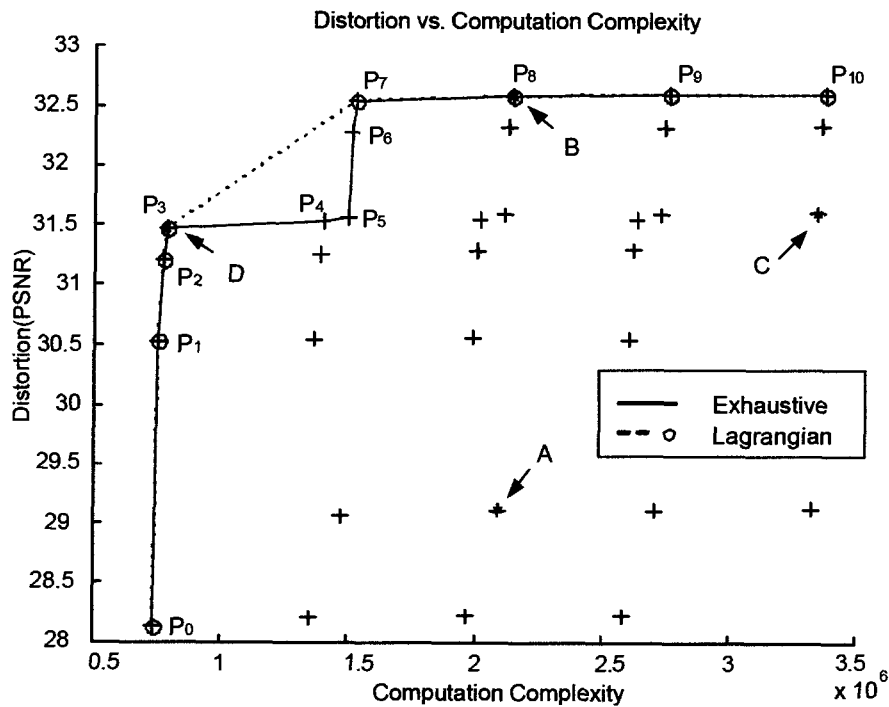
where MSD is an acronym of Mean Squared Difference and N is the number of pixels in the frame, and O_i and R_i are the intensity value of the original and the reconstructed frame. Note that the video coding system was set to the variable bit rate mode where its quantization parameter was fixed over the whole video sequence. The overall distortion data were measured in PSNR by averaging over 100 P-frames, using five video sequences, including *Carphone*, *Miss-America*, *Foreman*, *Salesman*, and *Claire*.

Table 5.4 Optimal operation modes found through the Lagrangian Method, where the given computational complexity is controlled by the Lagrangian multiplier λ over C-D data

Lagrangian Multiplier, λ	Operation Mode		
	ME s_1	I/H s_2	DCT s_3
0.000	3	1	3
0.001	2	1	3
0.009	1	1	3
0.020	0	1	3
0.454	0	0	3
7.434	0	0	2
18.350	0	0	1
77.187(∞)	0	0	0

5.4 EXPERIMENTAL RESULTS

Based on the data in Table 5.3, we searched optimal operating modes. Given the computational constraints C_{\max} , we were able to find optimal operating points by solving the optimization problem given in equations (5.2) and (5.3). We used two approaches, exhaustive search and the Lagrangian multiplier method. Note that our goal here was to find control variables s_1 , s_2 , and s_3 , to maximize the cost function of the optimization problem, since we dealt with the overall distortion in PSNR.



(a) Optimal operating modes

Let P_i , $i = 0, \dots, N-1$ represent an optimal operating point where N is the number of total optimal points by a search process. Using an exhaustive search, 11 optimal operating points were found and identified by P_{10} to P_0 in Figure 5.5(a). Their control parameters are same as follows: (3 1 3), (2 1 3), (1 1 3), (0 1 3), (0 1 2), (0 1 1), (1 0 3), (0 0 3), (0 0 2), (0 0 1), (0 0 0) respectively. However, as shown in Table 5.4, the Lagrangian method, detected only 8 optimal operating points. Optimal operating points not located on the convex hull curve are not detected [28]. This is shown graphically in Figure 5.5(a), where optimal operating points are drawn with a solid line, and a dotted line corresponds to an exhaustive search and the Lagrangian multiplier method, respectively.

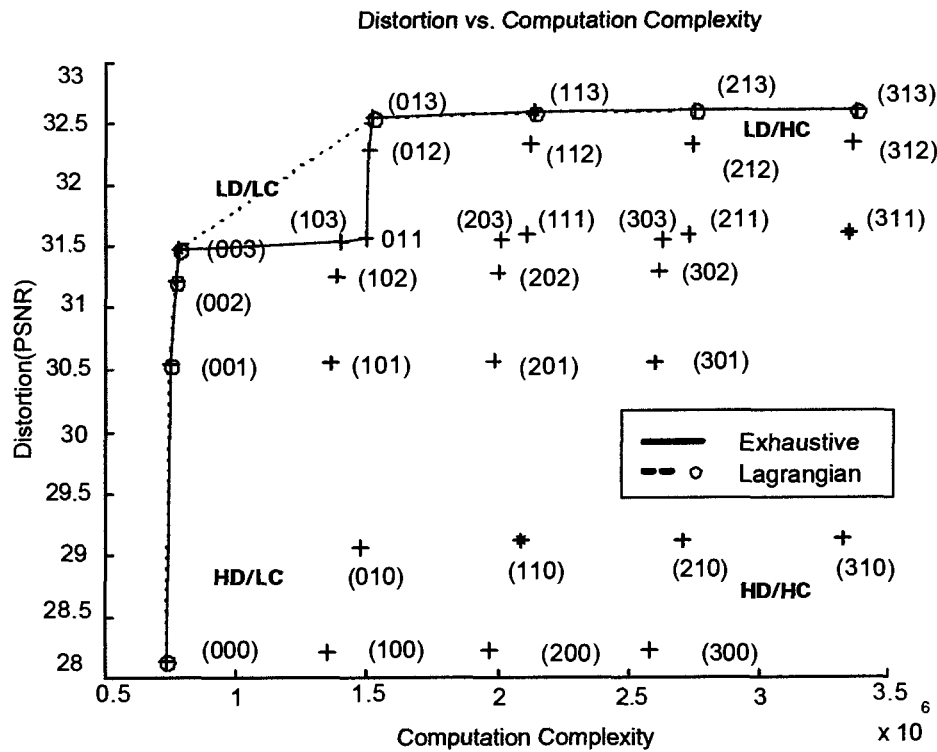
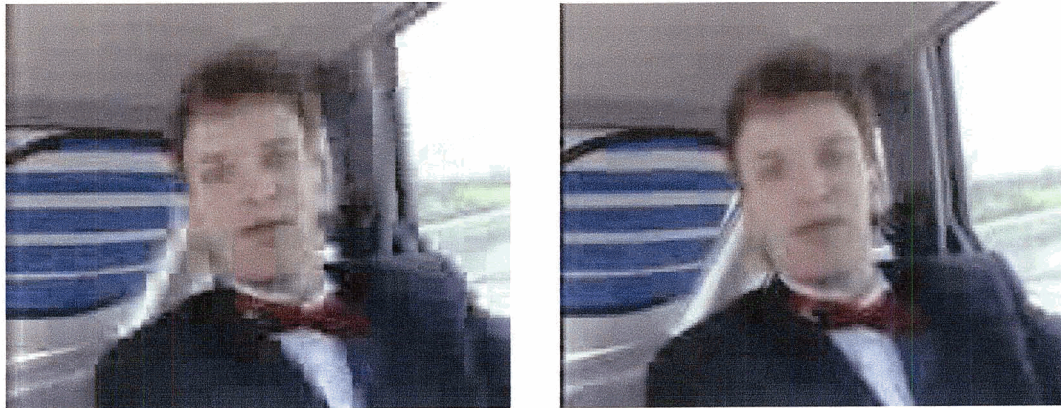


Figure 5.5 Optimal operating modes found through exhaustive search over the real-measured C-D (PSNR) data with test video sequences

Figure 5.5 also demonstrates how important it is, from an overall system performance point of view, to select optimal operating modes among control variables. Note that four operating modes A, B, C, and D are identified using the marker “*” in the figure, whose control parameters are respectively given as follows: (1, 1, 0), (1, 1, 3), (3, 1, 1), and (0, 0, 3). Operating modes C(3, 1, 1) and D(0, 0, 3) have similar average PSNR distortions, but significant difference in complexities requiring 3.3×10^6 and 0.8×10^6 operations, respectively.



(a) Mode A

(b) Mode B

Figure 5.6 Comparison in subjective quality for two modes, A and B of Figure 5.5 requiring similar computational complexity: the 6th frame, Inter coding, and QP=13 in the sequence *Carphone*

Operating modes A(1, 1, 0) and B(1, 1, 3) have similar complexities concerning 2.1×10^6 operations, but a 3.48dB difference in PSNR performance. This indicates that more computations do not necessarily perform better in an overall computation complexity space, which consists of combinations of all individual control variables. As expected, selecting optimal values of the control variables significantly influences the system's overall performance.

To demonstrate a comparison in the subjective performance, two sample video clips are shown in Figure 5.6, where the subjective quality is clearly distinct between two operating modes, A(1, 1, 0) and B(1, 1, 3) of Figure 5.5(a), closely located about 2.1×10^6 in the complexity axis. From this example, it is evident that the C-D optimal mode decision significantly affected the subjective performance of the video coding system.

In Figure 5(b), there are four regions classified according to the complexity and the distortion as follows: HD/LC(high distortion and low complexity), HD/HC(high distortion and high complexity), LD/LC(low distortion and low complexity), and LD/HC(low distortion and high complexity). As shown in the figure, two regions HD/LC and LD/LC require low complexities and locate down and up in the left. On the other hand, HD/HC and LD/HC require high complexity and locate up and down in the right respectively. Looking into the control parameters of modes and comparing one another located in different regions, it turns out that ME significantly influences the overall complexity, while DCT and H/I influence the overall distortion more than ME relatively.

Adaptive Mode Control

Video sequences have variations in characteristics including motion. This means that optimal operating modes defined by coding parameters change along with the changing video sequence. In other words, optimal C-D points should be controlled adaptively to achieve better performance. The adaptive control approach in regard to the operating modes is implemented and compared to the fixed approach. For the fixed method in the operating model control, the optimal control parameters given by (s_1, s_2, s_3) are searched in the initialization of the video encoding, under the given computational constraint, C_{\max} . These selected control parameters are used for all video frames and there is no update of the control parameters through whole video sequences.

For the adaptive approach, however, the optimal control parameters $(s_1, s_2, s_3)_{t+1}$ for the next frame $t+1$ are searched iteratively after encoding every frame based on the C-D data, whose data entry is updated with the distortion of control parameters $(s_1, s_2, s_3)_t$ at the current frame t .

Table 5.5 Performance comparison between the fixed and the adaptive control of the operating point (s_1, s_2, s_3) , with video sequences used in the model estimation

Constraint Control variable $\rho = 0.8$, $C_{\max} = 2701908$	Complexity (Instructions)		Distortion (PSNR)		Rate (Bits)	
	Fixed $(s_1, s_2, s_3) = (1, 1, 3)$	Adaptive	Fixed	Adaptive	Fixed	Adaptive
Carphone	2143449	2043211(0.95)	31.74	31.71	1908	1924(1.01)
Miss-America	2143449	2143449(1.00)	35.86	35.86	674	674(1.00)
Foreman	2143449	1917978(0.89)	30.49	30.43	2704	2717(1.01)
Salesman	2143449	1901267(0.89)	30.23	30.18	948	958(1.01)
Claire	2143449	2143449(1.00)	34.61	34.61	665	665(1.00)

Basically, this adaptive scheme arises from the fact that the frame distortion varies through the entire video sequence. The update equation for the new optimal mode in the adaptive approach is given below:

$$(s_1, s_2, s_3)_{t+1} = \min_{[s_1, s_2, s_3] \in \mathbb{S}_1 \times \mathbb{S}_2 \times \mathbb{S}_3} D_t(s_1, s_2, s_3) \quad (5.11)$$

subject to $C(s_1, s_2, s_3) \leq C_{\max}$.

where $(s_1, s_2, s_3)_{t+1}$ are the optimal control parameters for the frame $t+1$ and $D_t(s_1, s_2, s_3)$ is the distortion data in the C-D table, whose data entry is updated using the distortion of control parameters $(s_1, s_2, s_3)_t$ at the current frame t . In more detail, the algorithm of the adaptive mode control is described in the following steps.

Step 1: Let the computational constraint C_{\max} be given, and $(s_1, s_2, s_3)_0 = (2, 1, 2)$ is set for the I-frame coding in the first frame. Assume that the initial C-D data table, as given in Table 5.3, is available by pre-processing off-line.

Step 2: Encode the first frame in the I-frame mode using control parameters initially given $(s_1, s_2, s_3)_0 = (2, 1, 2)$.

Step 3: Optimal control parameters $(s_1, s_2, s_3)_t$ for frame t are searched from the C-D table. Encode in P-frame mode from the second frames.

Step 4: Calculate the distortion of $D_t(s_1, s_2, s_3)$ at the frame t corresponding to the control parameters $(s_1, s_2, s_3)_t$. Update the C-D table entry with the distortion $D_t(s_1, s_2, s_3)$.

Step 5: Increase the frame number $t = t + 1$ and jump back to Step 3. Repeat Step 3 to Step 5 until the end of sequence.

In following comparisons of rate performance, the video coding system was set to the variable bit rate mode, where its quantization parameter was fixed over whole video sequence, since the distortion model parameters were estimated with the fixed quantization parameter. Table 5.5 shows experimental results with the fixed and the adaptive control of operating modes. The same five video sequences involved in the

estimation process of the distortion parameters in the C-D model were used for the experiment. All 100 frames were coded and averaged, where the first frame was intra-coded and other following frames were inter-coded with the quantization parameter QP set to 13.

Let a variable $\rho \in \{0.0, \dots, 1.0\}$ denote a weighting factor to the computation complexity of the system represented by the maximum values of operation modes. The computational constraint value C_{\max} is relative to the maximum system complexity and derived by multiplying it with the constraint control variable ρ . It is shown in the table that C_{\max} is controlled by the constraint control variable ρ . This can be calculated by multiplying the control variable ρ to the maximum complexity of the operation mode, (s_1, s_2, s_3) , in the C-D model. This calculation can be given as

$$C_{\max} = \rho \times C(s_{1m}, s_{2m}, s_{3m}) \quad (5.12)$$

where ρ is the constraint control variable and $C(s_{1m}, s_{2m}, s_{3m})$ is the complexity for the operating mode (s_{1m}, s_{2m}, s_{3m}) , having the maximum complexity in the C-D model. The maximal complexity mode (s_{1m}, s_{2m}, s_{3m}) corresponds to (3, 1, 3) in the C-D model shown in Table 5.3. In Table 5.5, as an example, the constraint control variable was set to $\rho = 0.8$.

It is clearly proven in the table that the adaptive control works better with an active sequence, having more motions than with other silent sequences. For example, *Carphone*, *Foreman*, and *Salesman* sequences showed better performance with an adaptive control feature, while other silent sequences such as *Miss-America* and *Claire* showed no significant difference between the fixed and the adaptive control methods.

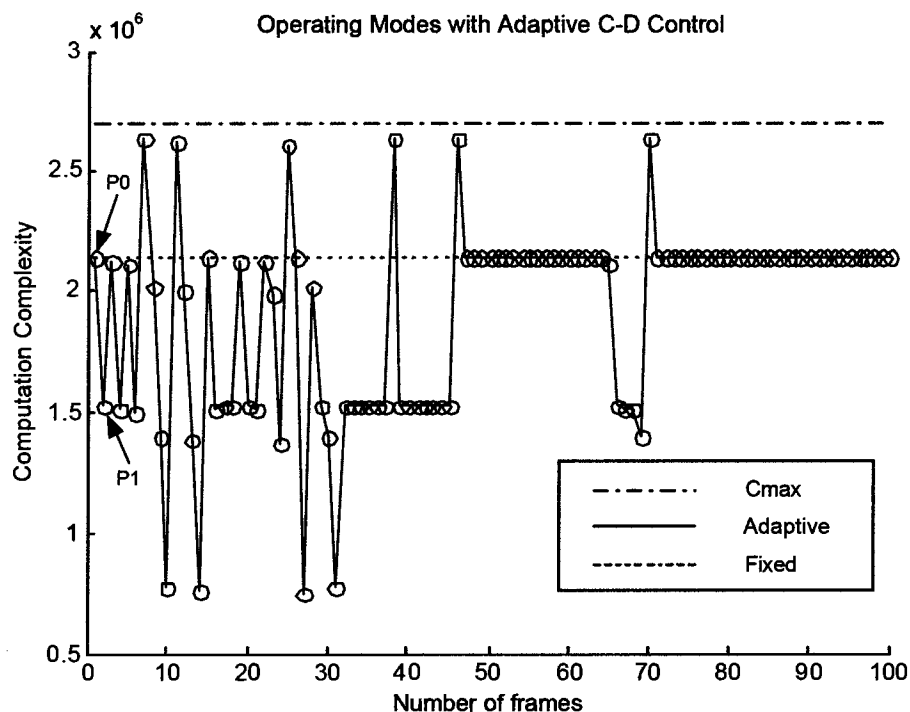


Figure 5.7 Operating mode found by adaptive C-D control in the sequence *Forman*

With the sequences *Forman* and *Salesman*, the computational complexity saved about 11% using the adaptive control, while it incurs degradation, less than 0.06dB. We also investigated how C-D optimization methods affect total bit rates. Generally, the bit rate is related to the coding efficiency, including motion estimation. As shown in the table, there is no significant difference of bit rate between the two control modes. Figure 5.7 shows complexity changes according to the operating modes detected adaptively by the C-D optimization algorithm.

Table 5.6 Performance comparison between the fixed and the adaptive control in the operating point (s_1, s_2, s_3) , with other video sequences not used in the model estimation

Constraint Control Factor	Complexity (Instructions)		Distortion (PSNR)		Rate (Bits)	
	Fixed $(s_1, s_2, s_3) =$ $(1, 1, 3)$	Adaptive	Fixed	Adaptive	Fixed	Adaptive
$\rho = 0.8,$ $C_{\max} =$ 2701908						
Container	2143449	1820380(0.85)	30.93	30.90	991	990(1.00)
Grandma	2143449	2009672(0.94)	32.02	32.01	582	577(0.99)
Mothr-Dautr	2143449	2068504(0.97)	31.50	31.49	1335	1338(1.00)
News	2143449	1742469(0.81)	30.79	30.75	1508	1545(1.03)
Suzie	2143449	2124939(0.99)	33.02	33.02	1664	1667(1.00)
Flower-Garden	2143449	1822708(0.85)	26.40	25.73	17983	17364(0.97)

In the figure, operating modes P_i , $i \in \{0, \dots, N-1\}$ are represented with the control parameters (s_1, s_2, s_3) . Complexity numbers corresponding to the operating modes are the same as ones shown in TABLE III. For example, the first 10 operating modes P_i , $i \in \{0, \dots, 9\}$ are given as follow respectively: $(1, 1, 3)$, $(0, 1, 3)$, $(1, 1, 2)$, $(0, 1, 2)$,

(1, 1, 1) , (0, 1, 1) , (3, 0, 3) , (2, 0, 3) , (1, 0, 3) , (0, 0, 3) , (3, 0, 2) , (2, 0, 2) ,
 (1, 0, 2) , (0, 0, 2) .

Note that the distortion parameters of the C-D model were estimated using five video sequences. It would be interesting to investigate how much more effective the estimated model parameters would be with other video sequences not involved in the model estimation process.

Table 5.6 shows experimental results using the following five video sequences: *Container*, *Grandma*, *Mothr-Dautr*, *News*, and *Suzie*. The quantization parameter QP was fixed to 13. The first frame was intra-coded and those that followed were inter-frame coded. For the sake of comparison, the results were obtained by averaging over 100 frames. As shown in the table below, the C-D model works well, even with other video sequences not considered in the model estimation process. With active sequences such as *Container* and *News*, the adaptive control method performed best in the C-D optimization.

With the various sequences above, computation reductions were obtained up to 19% compared to the fixed method, while the degradations of the reconstructed video were less than 0.05dB. Furthermore, there was no significant difference between the adaptive and the fixed methods in rate performance. Based on these experimental results, it is evident that the estimated C-D model parameters are accurate enough to be applied to most video sequences, regardless of their motion.

5.5 SUMMARY

The performance of a computationally configurable video coding scheme with respect to computational complexity and distortion, has been analyzed. The proposed coding scheme consists of three coding modules: motion estimation, sub-pixel accuracy, and

DCT pruning, whose control variables can take several values, leading to significantly different performance for the coding. This analysis confirms that a configurable video coding system where the control parameters are chosen optimally leads to better performance. To evaluate the performance of proposed scheme according to input video sequences, we applied video sequences other than those involved in the process of model parameter estimation, and showed that the model parameters are accurate enough to be applied regardless of the type of input video sequences. Furthermore, an adaptive scheme to find the optimal control parameters of the video modules was introduced and compared with the fixed. The adaptive approach was proven to be more effective with active video sequences rather than with silent video sequences.

Chapter 6

CONCLUSION

As a solution to alleviate the computational requirements of the motion estimation algorithm, a fast and efficient scheme [70], based on a 1-D gradient fast search that reduces the probability of being trapped in a local minimum, was introduced and evaluated in its search speed and motion estimation performance. Basically, the proposed method can be applied to other fast-search methods as well. Especially, its performance improvement can be traded-off with computation cost, according to application requirements.

Furthermore, two fast half-pel search methods [69, 72] were developed. One [72] is based on an approximate model of the error-criterion function and was presented in the previous chapter. The pre-computed error-criterion values being computed at full-pixel level are used to derive the motion vector and the error-criterion values at sub-pixel accuracy. Hence, the approach reduces dramatically the number of computations compared to conventional methods, where the error-criterion function at the sub-pixel accuracy is computed directly from interpolated sub-pixel values. The other method uses an efficient search pattern, proposed in [69], which reduces the computational complexity to 50% of that of the conventional method. As a matter of fact, the computational complexity of the half-pel search is comparable to that of the integer-pel search when a fast search algorithm is applied to video coding. In other words, the half-pixel accuracy motion estimation module has a significant role in improving the whole video-coding speed, especially with moderate motion video sequences since it takes more processing power than the integer-pixel accuracy search under the fast-search framework. In

particular, the proposed method is viable in real-time video coding such as video-telephony and videoconferencing, where a slight degradation of video quality could be allowed in a trade-off with video-coding speed.

A fast and efficient approach to rate-distortion optimization was introduced, based on an adaptive rate-distortion model, and which subsequently reduced prohibitively extensive computation in comparison with traditional approaches. It was shown that overall rate and distortion performance could be improved close to that of the optimal algorithm by choosing the optimal quantization parameter and motion vector that minimize residual bit-rates and distortion through a fast rate distortion optimization algorithm. The parametric-approximation model of rate and distortion function consisting of quantization parameter was utilized for the estimation of real rate and distortion value. This resulted in the substantial reduction of computational complexity associated with DCT and quantization, while it incurred a small sacrifice in rate-distortion performance. In order to verify the performance of the proposed adaptive model, the rate and distortion optimization was conducted with regard to the quantization parameter in the frame layer. In future work, it will be extended to the macroblock layer optimization by considering motion vector, and coding mode as well.

A scalable coding scheme [65] capable of optimally selecting a coding parameter through a systematic method was introduced so that the system could obtain the best performance under the given computational constraints. First, major coding modules were identified and analyzed in terms of computational complexity and distortion in the H.263 video coding framework. When a control parameter was chosen, its deterministic property was taken into account so that the system could achieve hard control over the overall computational complexity. Based on the analyzed C-D data, the operational C-D curve was driven through an exhaustive search and the Lagrangian optimization. The efficiency of optimal operational modes were confirmed using test video sequences, by showing how closely the operational C-D curve obtained from the analyzed data for each coding

modules approximates the overall C-D curve obtained through direct measurement over the whole video sequence. It was proven that an optimally chosen operational mode makes a significant difference, compared to that of the worst mode under the given computational constraints. Moreover, an adaptive scheme was carried on updating the C-D data so that the system could make a good track of the optimal operational modes, which vary according to the input video property.

As a summary, all research works carried out and presented in the thesis were driven in real-time coding and low bit rate (LBR) application perspective. They covered error concealments techniques over error prone wireless channels, fast and efficient motion estimation methods, rate and distortion optimization between motion estimation and error residual coding, a configurable video coding framework to optimally control the complexity of coding system for the best performance in terms of PSNR. For a future work, the configurable framework would be further extended by taking into accounts most coding modules in real video system. The fast and efficient motion estimation techniques would be considered for their VLSI architecture design in low-powered mobile applications.

BIBLIOGRAPHY

- [1] A. Ortega, K. Ramchandran, "Rate distortion methods for image and video compression", *IEEE Signal Processing Magazine*, Nov. 1998
- [2] G. J. Sullivan, T. Wiegand, "Rate distortion optimization for video compression", *IEEE Signal Processing Magazine*, Nov. 1998
- [3] B. Girod, "Rate constrained motion estimation", in *Proc. Conf. Visual Commun. Image Processing*, Vol. 2308, SPIE, 1994, pp. 1026-1034
- [4] G. M. Schuster, A. K. Katsaggeelos, "Fast efficient mode and quantizer selection in the rate distortion send for H.263", in *Proc. Conf. Visual Commun. Image Processing*, SPIE, Mar. 1996, pp. 784-795
- [5] K. Lengwehasatit, A. Ortega, "Rate complexity distortion optimization for quadtree based DCT", *Image Processing 2000*.
- [6] V. Goyal, M. Vetterli, "Computation distortion characteristics of block transform coding" in *Proc. of ICASSP'97*, Munich, Germany, Apr. 1997
- [7] I. Ismaeil, A. Docef, F. Kossentini, and R. Kreidieh, "A computation-distortion optimized framework for efficient DCT-based video coding", *IEEE Trans. Multimedia*, Vol. 3, No. 3, Sept. 2001
- [8] ITU-T Study Group 15, Draft Recommendation H.263, Apr. 7, 1995
- [9] J. R. Jain, A.K. Jain, "Displacement measurement and its application in interframe image coding", *IEEE Trans. Commun.*, Vol. COM-29, pp.1799-1808, Dec. 1981
- [10] 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
- [11] J. Y. Tham, S. Ranganath, M. Ranganath, A. A. Kassim, "A novel unrestricted center-biased diamond search algorithm for block motion estimation", *IEEE Trans. Circ. and Syst. for Video Technol.*, Vol. 8, No. 4, Aug. 1998

- [12] S. Zhu, K. K. Ma, "A new diamond search algorithm for fast block-matching motion estimation", *IEEE Trans. On Image Processing*, Vol. 9, No. 2, Feb. 2000
- [13] B. Girod, "Motion-compensating prediction with fractional-pel accuracy", *IEEE Trans. Commun.*, Vol. 41, No. 4, Apr. 1993
- [14] V. Bhaskaran, K. Konstantinides, *Image and video compression standards : algorithms and architectures*, Second edition, Kluwer Academic, 1997.
- [15] H. Fujiwar, "An all-ASIC implementation of a low bit-rate video codec", *IEEE Trans. On Circuit and Systems for Video Technology*, June, 1992
- [16] K. Gutttag, R.J. Cove, and J.R. Van Aken, " A single chip multiprocessor for multimedia:the MVP", *IEEE Computer Graphics and Applications*, Nov. 1992.
- [17] C. G. Zhou, "MPEG video decoding with the UltraSPARC visual instruction set", *IEEE Digest of Papers COMPCON Spring 1995*, March 1995
- [18] B. Furht, J. Greenberg, R. Westwater, *Motion estimation algorithms for video compression*, Kluwer Academic Press, 1997
- [19] P. Kuhn, *Algorithms, complexity analysis and VLSI architectures for MPEG4 motion estimation*, Kluwer Academic Press, 1999
- [20] B. G. Lee, "A new algorithm to compute the discrete cosine transform", *IEEE Trans. On ASSP* Dec. 1984
- [21] K. R. Rao, P. Yip, "Discrete cosine transform – algorithms, advantages, applications", Academic Press, 1990
- [22] Y. Arai, T. Agui, and M. Nakajima, "A fast DCT-SQ scheme for images", *Transactions of the IEICE*, E-71(11):1095-1097, Nov. 1988
- [23] A. N. Skodras, "Fast discrete cosine transform pruning", *IEEE Trans. on Signal Processing*, Vol. 42, No. 7, July 1994
- [24] Z. Wang, "Pruning the fast discrete cosine transform", *IEEE Trans. On Comm.* Vol. 39, No. 5, May 1991
- [25] S. C. Chan, K. L. Ho, "A new two-dimensional fast cosine transform algorithm", *IEEE Trans. on Signal Processing*, Vol. 39, NO. 2, pp. 481-485

- [26] G. M. Schuster, A. K. Katsaggelos, "A theory for the optimal bit allocation between displacement vector field and displaced frame difference", *IEEE Journal on Selected Areas in COMM.* Vol. 15, No. 9, Dec. 1997
- [27] Y. Yang, S. S. Hemami., "Generalized rate distortion optimization for motion compensated video coders", *IEEE Trans. On Circuits and Systems for Video Technology*, VOL. 10, NO. 6, SEPT. 2000
- [28] G. M. Schuster, Aggelos K. Katsaggelos, *Rate distortion based video compression*, Kluwer Academic Publishers, 1997
- [29] C. Y. Hsu, A. Ortega, "A Lagrangian optimization approach to rate control for delay-constrained video transmission over burst error channels", in *Proc. of ICASSP'98*, (Seattle WA), May, 1998
- [30] A. Ortega, "Optimal bit allocation under multiple rate constraints", in *Proc. Data Compression Conference*, Snowbird, UT, April, 1996
- [31] J. J. Chen, D. W. Lin, "Optimal bit allocation for video coding under multiple constraints", in *Proc. IEEE Intl. Conf. On Image Proc.*, ICIP'96, 1996
- [32] K. Ramchandran, M. Vetterli, "Best wavelet packet bases in a rate distortion sense", *IEEE Trans. on Image Proc.*, Vol. 2, pp. 160-175, Apr. 1993
- [33] Y. Shoham, A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers", *IEEE Trans. ASSP*, Vol. 36, pp. 1445-1453, Sep. 1988
- [34] G. M. Schuster, A. K. Katsaggelos, "An optimal quad tree based motion estimation and motion based interpolation scheme for video compression", *IEEE Trans. on Image Proc.*, Vol. 7, No. 11, pp. 1505-1523, Nov. 1998
- [35] C. E. Shannon, "A mathematical theory of communications", *Bell system tech. journal*, 27:397-423, 1948
- [36] Mohammed Ghanbari, *Video coding: an introduction to standard codecs*, The Institute of Electrical Engineers, 1999
- [37] "Video Codec for Audiovisual Services at px64kbits", *ITU-T Recommendation H.261*, 1993

- [38] ISO/IEC, "Video, coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s", 1991
- [39] ISO/IEC, "Generic coding of moving pictures and associated audio information: Video", 1995
- [40] ISO/IEC, "Visual, Information Technology - Coding of audio visual objects", 1999
- [41] K. H. lee, J. H. Choi, B. K. Lee and D. G. Kim, "Fast two step half-pixel accuracy motion vector prediction", *Electronics Letters*, 30th Mar., 2000 Vol. 36, No. 7
- [42] R. Srinivasan, K.R. Rao, "Predictive coding based on efficient motion estimation", *IEEE Trans. Commun.*, Vol. Com-33, No.8, Aug. 1985
- [43] M. J. Chen, L. G. Chen, T. D. Chiueh, "One-dimensional full search motion estimation algorithm for video coding", *IEEE Trans. Cir. and Syst. for Video Technol.*, Vol. 4, No. 5, Oct. 1994
- [44] O. T. Chen, "Motion estimation using a one-dimensional gradient descent search", *IEEE Trans. Circ. and Syst. for Video Technol.*, Vol. 10, No. 4, Jun. 2000
- [45] M. Gallant, G. Cote, F. Kossentini, "An efficient computation constrained block-based motion estimation algorithm for low bit rate video coding", *IEEE Trans. On Image Processing*, Vol. 8, No. 12, Dec. 1999
- [46] I. Ismaeil, A. Docef, F. Kossentini, R. Ward, "Efficient motion estimation using spatial and temporal motion vector prediction", *ICIP 99*, Vol. 1, 1999
- [47] F. Kossentini, Y. Lee, "Computation-constrained fast MPEG-2 encoding", *Signal Processing Lett.*, Vol.4, pp.224-226, Aug. 1997
- [48] C. H. Hsieh, P. C. Lu, J. S. Shyn, "Motion estimation using inter-block correlation", *IEEE International Symposium on Circ. and Syst.* Vol. 2, 1990
- [49] L. G. Cheh, W. T. Chen, Y. S. Jehng, T. D. Chiueh, " A predictive parallel motion estimation algorithm for digital image processing", *ICCD 1991*, pp. 617-620, 1991
- [50] A. N. Netravali, B. G. Haskell, "Digital pictures: representation and compression", 1988, Plenum Press
- [51] M. Wada, "Selective recovery of video packet loss using error concealment", *IEEE Journal on Selected Areas in Comm.*, Vol. 7, No. 5, June '89

- [52] Y. Wang, Q. Zhu, "Error control and concealment for video communications: a review", *Proceedings of IEEE*, Vol.86, No.5, May '98
- [53] J. I. Ronda, M. Eckert, F. Jaureguizar, and N. Garcia, "Rate control and bit allocation for MPEG-4", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 9, No. 8, Dec. 1999
- [54] T. Chiand and Y. Q. Zhang, "A new rate control scheme using quadratic rate distortion model", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 7, No. 1, Feb. 1997
- [55] L. Lin, A. Ortega, and C. Kuo, "Cubic spline approximation of rate and distortion function for MPEG video", in *Proc. of the SPIE*, Vol. 2668, Jan. 1996, pp. 169-180
- [56] W. C. Chung, F. Kossentini, M.J.T. Smith, "An efficient motion estimation technique based on a rate-distortion criterion", *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., Volume: 4*, 1996
- [57] S. Y. Hu, M. C. Chen and N. W. Jr, "A fast rate-distortion optimization algorithm for motion-compensated video coding", *IEEE International symposium on circuit and systems*, Jun. 1997, Hong Kong
- [58] L. C. Hamilton, *Regression with Graphics*, Duxbury Press, 1992
- [59] R. F. Gunst, R. L. Mason, *Regression analysis and its application*, Marcel Dekker, Inc., New York and Basel, 1980
- [60] M. C. Chen, A. N. Willson, Jr., "Rate-distortion optimal motion estimation algorithm for video coding", *IEEE international Conference on Acoustics, Speech, and Signal Processing, ICASSP-96*, Vol. 4, pp 2096-2099, 1996
- [61] M. C. Chen, A. N. Willson, Jr., "Rate-distortion optimal motion estimation algorithms for motion compensated transform video coding", *IEEE Trans. on Circuit and Systems for Video Technology*, Vol. 8, No. 2, April 1998
- [62] M. Z. Coban, R. M. Mersereau, "A fast exhaustive search algorithm for rate constrained motion estimation", *IEEE Trans. on Image Processing*, Vol. 7, No. 5, May 1998

- [63] B. Tao, B. W. Dickinson, H. A. Peterson, "Adaptive model-driven bit allocation for MPEG video coding", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 10, No. 1, Feb. 2000
- [64] N. Jayant and P. Noll, *Digital coding of waveforms*. Englewood Cliffs, NJ: Prentice-Hall, 1984
- [65] D. N. Kwon, "Computation complexity and performance optimization in video coding system", *IEEE Wire and Wireless Network Conference 2003*
- [66] J. Jung, W. Ahn, "Sub-pixel accuracy motion estimation algorithm using a model for motion compensated errors", *PCS93*, 1993
- [67] Y. Senda, H. Harasaki, M. Yano, "Theoretical background and improvement for a simplified half-pel motion estimation", *International Conference on Image Processing*, Vol. 3, 16-19 Sept. 1996
- [68] X. Li, X. Gonzales, "Locally quadratic model of the motion estimation error criterion function and its application to sub-pixel interpolation", *IEEE Trans. On CSVT*, Vol. 6, No. 1, Feb. 1996
- [69] D. N. Kwon, "Half-pixel accuracy fast search in video coding", *IEEE ISSPA 2003*
- [70] D. N. Kwon and P. Driessen, "Efficient and fast predictive motion estimation algorithm for low bit rate video coding", *IEEE PACRIM01*, Aug, 2001
- [71] D. Kwon and P. Driessen, "Error concealment techniques for H.263 video transmission", *IEEE PACRIM99*, Aug, 1999
- [72] D. N. Kwon, "Sub-pixel accuracy motion estimation algorithm using a linear approximate model of the error criterion function", submitted to *IEEE Transactions on Multimedia*