POWER AND DISTORTION OPTIMIZED VIDEO CODING FOR PERVASIVE COMPUTING APPLICATIONS

by

YONGFANG LIANG

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON

May 2006

ACKNOWLEDGEMENTS

Pursuing a doctoral degree is a long journey that consists of both the bitterness of failure and the joy of success. I was lucky to have trusty, worthy and warmhearted people around me on this journey. I would not have accomplished my goals without all their help and support. One great thanks to you all!

I am very grateful to my advisor and committee chair, Professor Ishfaq Ahmad for his guidance, trust and continuous support throughout the past four years. I cannot imagine accomplishing what I have without the benefit of his vast knowledge, tremendous and insightful guidance, and words of encouragement along the way. Without his persistent help this dissertation would not have been possible.

I am also grateful to my dissertation committee members, Dr. Alp Aslandogan, Dr. Hua-Mei Chen, Dr. Jean Gao, and Dr. Jung-Hwan Oh, for sharing their ideas and support with me over the past four years. My special thanks go to Dr. Zhihai He for sharing his research ideas with me. I also appreciate the assistance of those who gave me their help and support in these years.

A personal thank you to my family and relatives, especially my parents, Mingyuan Liang and Zuanxi Li, for their care, encouragement and sacrifice for the past many years. Finally, I specially thank my beloved wife, Limei Mai, who has been at my side, loving and supporting me unconditionally in all these years.

April 8, 2006

ABSTRACT

POWER AND DISTORTION OPTIMIZED VIDEO CODING FOR PERVASIVE COMPUTING APPLICATIONS

Publication No. _____

YONGFANG LIANG, PhD.

The University of Texas at Arlington, 2006

Supervising Professor: Ishfaq Ahmad

This dissertation investigates video encoding schemes for pervasive computing applications that must ensure low power consumption in addition to high compression efficiency. The contribution of the dissertation is the formulation of a theoretical problem that captures the joint optimization of power and distortion in video coding. The study of the complexity distribution of typical video encoders helps to develop a complexity-scalable video encoding architecture that includes several control parameters to adjust the power consumption of the major modules of the encoder. An analytic framework to model, control and optimize the power-rate-distortion is developed, which facilitates the development of optimization schemes to determine the best configuration of the complexity control parameters according to the either or both

the power supply level of the device and the video presentation quality. The dissertation proposes complexity control schemes that dynamically adjust the control parameters. Using extensive simulations on an instruction set simulator, the accuracy of the model, and quality of the optimization schemes are investigated. For additional performance improvement, we propose algorithms that exploit the video content to reduce the power consumption and improve the video quality. This is done by obtaining and maintaining the "motion history" of a video sequence in a hierarchical fashion. By adaptively adjusting the complexity parameters according to the motion history information gained from the video sequence, the power is saved when the scene has little motion and consumed when the motion activity increases. Extensive experiments have been performed to show the validity and merits of the proposed techniques.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Ubiquitous Video Coding for Pervasive Computing Applications

Video coding/compression is the key ingredient for cost-effective storage and transmission of video images over any digital communication link. There has been a progression over time in the study of video coding technologies. This progression has gone from video image difference coding, motion-compensated hybrid coding, to object-based coding. As the multimedia applications are becoming more sophisticated and complex, video coding technologies have to evolve in order to meet the ever-increasing processing requirements.

1.1.1 Digital Video Compression Overview

A raw video stream tends to be quite demanding when it comes to storage requirements, and demands for high network capacity when being transferred. For example, an uncompressed HDTV picture with 2.2 million pixels and raw coding with 24 bits per pixel (8 per color component) would require 1.5-3 Gbits/sec depending on the picture frequency. Therefore before being stored or transferred, the raw stream is usually transformed to a representation using compression. The purpose of video coding is to enable the transmission and storage of video in digital form with as small a bandwidth and as good a quality as possible. In the past several decades, the power of

computers and the advancement of mathematical algorithms for video compression have lead to a tremendous success in a diversity of video applications, such as multimedia authoring, entertainment, digital CATV, video conferencing, video on demand and digital direct broadcast satellite TV.

Table 1.1 Video Codecs in various video services and applications

Services and applications		Video Codec
3GPP (3rd generation Partnership Project)	MMS	H.263 profile 0 level 10
	PSS Release 5	H.263 profile 0 level 10
	PSS Release 6	N/A
3GPP2	Circuit Switched Video Conferencing Services	MPEG-4 Simple Profile Level 0 and H.263 baseline
	Packet Switched Video Conferencing Services	MPEG-4 Visual or ITU-T H.263 (or both) shall be supported
3G-324M mobile videoconferencing		H.263 baseline level 10
H.320 videoconferencing		H.261 QCIF
H.323 videoconferencing		H.261 QCIF (if have video service)
HD-DVD, Blue-ray DVD		Microsoft VC-9 (VC-1), MPEG-2, MPEG-4 AVC (H.264)
DVD		MPEG-2 Main Profile @ Main Level
SVCD (Super Video CD)		MPEG-2 MP @ Low Level MPEG-1
VCD (Video CD)		MPEG-1

To ensure interoperability between different terminal devices when sharing video content and compliance when creating (or personalizing) video bitstreams, different video coding standards/recommendations have been proposed targeting

various application areas, including H.261/H.263/H.264 by ITU-T [55][56][57] and MPEG-1/MPEG-2/MPEG-4 by ISO/IEC [51][52][53]. The H.26x recommendations have been designed mostly for real-time video communication applications, such as video conferencing and video telephony. On the other hand, the MPEGx standards are designed to address the needs of video storage (DVD), broadcast video (broadcast TV), and video streaming. These standards/recommendations have been the engines behind the commercial success of various video services and applications, summarized in Table 1.1, each of those was designed to fit a specific application and best suited to particular requirements.

MPEG-1 [51] is a standard for the compression of moving pictures and audio up to 1.5 Mbits/sec. This was based on CD-ROM video applications, and is a popular standard for video on the Internet, transmitted as .mpg files. MPEG-1 is the standard of compression for VideoCD, the most popular video distribution format throughout much of Asia. In addition, level 3 of MPEG-1 is the most popular standard for digital compression of audio--known as MP3. MPEG-2 is based on MPEG-1, but designed for the compression and transmission of digital broadcast television. The most significant enhancement from MPEG-1 is its ability to efficiently compress interlaced video. MPEG-2 [52] can be used for application between 1.5 and 15 Mbits/sec such as Digital Television set top boxes and DVD movies. MPEG-2 scales well to HDTV resolution and bit rates, obviating the need for an MPEG-3. The focus and scope of MPEG-4 [53] was defined as the intersection of the traditional separate industries of telecommunications, computer, and file where audio-visual applications exist. It aims at

application such as Internet and intranet video, video e-mail, home movies, virtual reality games, simulation and training. MPEG-4 addresses the need [32] for: (1) Universal accessibility and robustness in error prone environments. (2) High interactive functionality. (3) Coding of natural and synthetic data. (4) Compression efficiency. Besides the new functionalities, MPEG-4 is also supporting the basic functionalities such as synchronization of audio and video, low delay mode, coding of a variety of audio types, interoperability with other audio-visual systems, support for interactivity, ability to efficiently operate in the 9.6 to 1024 Kbit/s range, and ability to operate in different media environments.

H.261 [54] is an ITU standard designed for two-way communication over ISDN lines (video conferencing) and supports data rates that are multiples of 64Kbit/s. H.261 uses intraframe and interframe compression. The algorithm is based on Discrete Cosine Transform (DCT) and can be implemented in hardware or software. H.261 supports CIF and QCIF resolutions. H.263 is based on its predecessor H.261, using the same hybrid block-based coding structure, with enhancements that improve video quality over modems. H.263 [55] was developed for low bit rate video coding between 20 and 64kbps. H.263 supports CIF, QCIF, SQCIF, 4CIF and 16CIF resolutions. It has been widely used in videoconferencing and video-telephony applications. H.264 (also known as MPEG-4 AVC) [56] is the latest international video coding standard. It was jointly developed by the Video Coding Experts Group (VCEG) of the ITU-T and the Moving Picture Experts Group (MPEG) of ISO/IEC with an objective to create a single video-coding standard, which simultaneously resulted in AVC (Advanced Video coding) of

MPEG-4 Part 10 and ITU-T H.264 recommendation. While using the same hybrid block-based motion compensation and transform-coding model as H.263, MPEG-2 and MPEG-4, H.264/AVC achieves a significant improvement in compression efficiency relative to prior standards, by applying a number of new features and capabilities. It can be used in a wide range of applications, including video telephony, video conferencing, TV, storage (DVD and/or hard disk based, especially high-definition DVD).

1.1.2 Ubiquitous Video Coding

At the turn of the last century, we witnessed the genesis of the "wireless and pervasive computing era," in which the focus of telecommunications and computing started to change, from traditional wired telephony-oriented services and infrastructures to data-based services, and from desktop workstations to smart hand-held, personal digital assistants, and mobile computers [1]. With the advances in high-capacity memory storage, powerful microprocessor technologies and efficient video coding algorithms, pervasive video applications are likely to prevail in the next-generation wireless networks, providing exciting new services to users in a home or small office environment, such as mobile video conferencing or streaming on hand held devices like PDA (Personal Digital Assistants).

Ubiquitous video encoding is envisioned for a wide range of applications, such as battlefield intelligence, surveillance, reconnaissance, security monitoring, emergency response, disaster rescue, environmental tracking, tele-medicine, and multimedia systems in consumer electronics [2]. One application example of ubiquitous video coding is the development of an Aware Home [50], which makes video and audio

sensing transparent to everyday activities and distributing it everywhere. By ubiquitous video coding, the system allows the users to communicate within a house using a videophone like device, control devices in the home using voice or gestures, and monitor children and pets anywhere in the house. Another example is the wireless camera flashlight used by police officers as part of a pervasive video surveillance system. The officers can use the flashlight to capture the crime scene, process it and communicate with the department for the purpose of criminal identification or archiving.

While the video compression technology has matured in the realms of entertainment, television, and movie, current state-of-the-art is inadequate for the changing computing platforms and paradigms in the upcoming wireless and pervasive computing era. In ubiquitous video coding applications, video capture, compression and network streaming mostly operate on a device/sensor with limited energy. A primary factor in determining the utility or operational lifetime of the mobile communication device is how efficiently it manages its energy consumption. Traditional video coding technologies do not consider the power supply problem. Moreover, the coding environment is ubiquitous and the coding content may changes over time very frequently. These new video coding characteristics pose new research challenges to the researchers.

1.2 Research Challenges

Video compression technologies in futuristic environments will have to meet the following challenges: (a) Aware of the **P**ower of the underlying device; (b) Ensuring **Q**uality-of-Service; (c) Catering to **R**ough and un-ideal environments; (d) Fast encoding **S**peed. We collectively refer to these research goals as the **PQRS** goals, which serve as the motivation of the research in this dissertation:

- *Power Awareness*. In wireless pervasive video communication environments, one fundamental problem is how to efficiently manage the power consumption of the video encoding system, while preserving a well-perceived video presentation quality. The nickel-cadmium and lithium-ion batteries general used in mobile devices have increased their energy capacity roughly by 10 to 15 percent per year. However, it is conjectured that only another 15 to 25 percent increase is possible [74]. The implication of limited power on video encoding is two-fold. First, efficient video compression significantly reduces the amount of the video data to be transmitted, which, in turn, saves a significant amount of energy in data transmission. Second, more efficient video compression often requires higher computational complexity and larger power consumption in computing. To prolong the lifetime of the battery, a video encoding system capable of adjusting its energy consumption as demanded by the situation and its environment is highly desirable.

- *Ensuring Quality of Service*. Another challenge is to ensure quality-of-service (QoS), which in the context of video encoding is measured from two perspectives: First, the picture quality of motion picture must be optimized. Nothing

prevents a researcher to aim for better than HDTV quality on a handheld device, and the optimization pursuit may be without limit. Second, the available coding bits are best used when and where needed the most at limited transmission bandwidth.

- *Content Adaptive*. Pervasive environments are far from ideal. They contain highly obscure and unintelligible data, which exert tremendous stress on an encoder, in terms of the computational requirements as well as the bit budget to encode the scene complexity. Therefore, we also need to develop effective adjustment techniques to consume minimal power while preserving desired video quality, according to the changes of the environment as well as the power supply level.

- *Fast Encoding Speed*. Real-time video compression remains a computationally demanding problem. With limited CPU power available in pervasive computing devices, such as miniature cameras and sensors, another research challenge is to design fast algorithms without compromising on the solution and at the same time invent techniques to speed up the execution of the algorithms.

Unfortunately, not much research work and attention have been paid to the new requirements of video coding for pervasive computing applications, although some of these issues have been addressed separately in some related research areas with different goals, as reviewed in the following chapter. The focus of this dissertation is to make an attempt to meet the PQRS goals in ubiquitous video coding under power constraints, on techniques for efficiently utilizing the energy supply while preserving desirable video quality. We consider not only the video encoding architecture, but power and quality control schemes as well. Our intent is to develop video coding

technologies that sufficiently meet these new challenges, and to assess the best performance trade-offs as well as the collective impact of rate, distortion, and power.

CHAPTER 2

A SURVEY ON RELATED RESEARCH WORKS

2.1 Low Complexity Video Coding

The source coding schemes recommended by the international standardization groups are very sophisticated in order to achieve high bit rate reduction under the constraint of the highest possible picture quality. These coding schemes will result in systems of high complexity, which usually lead to high power consumption. Thus cost effective implementation of these high complexity systems is one of the most important criteria for many signal-processing system designs, particularly in pervasive computing applications such as multimedia cellular applications and multimedia system on chip design that have limited processing power. This is usually known as fast/real-time video encoding/decoding.

2.1.1 Low Level System Optimization

Low-level system optimization is an important technique to reduce the power consumption. There have been many approaches to achieve this design goal at implementation levels ranging from very-large-scale-integration (VLSI) fabrication technology to parallel processing.

2.1.1.1 VLSI Design

VLSI technologies have now advanced to the point where the processing power and memory required to perform real time video compression and decompression can

be incorporated into a few silicon chips. Many VLSI design architectures for fast video coding have been proposed in the literature. Research works of low-power VLSI design in video coding are reported in [25]. In [35], the authors present an architectural enhancement to reduce the power consumption during full-search block-matching motion estimation. The proposed approach achieves power savings by disabling portions of the architecture that perform unnecessary computations. [34] compares different full-search motion estimation architectures targeted for low-power consumption. Each of the architectures is analyzed, and then compared to the others. In [7], five low-level circuit designs for the DCT/IDCT operations in an H.263 system are investigated for low-power improvements. The techniques include skipping low energy DCT input, slupping all-zero IDCT input, low precision constant multipliers, clock gating, and a low transition data path. Another two instances of low-power VLSI design for motion estimation can be found in [44] and [30]. An overview on architectures for VLSI implementations of video compression schemes is presented in [75].

2.1.1.2 Parallel Processing

The latest developments in parallel and distributed systems promise a higher degree of performance at an affordable cost (such as a network of workstations or PC's). [42] describes a software-based MPEG-4 video encoder, which is implemented using parallel processing on a cluster of workstations collectively working as a virtual machine. Another example of using multiprocessor parallel systems is the implementation of hybrid encoding of CIF video in [38]. An overview of techniques to

implement various image and video compression algorithms using parallel processing can be found in [80].

2.1.2 High Level Optimization

A variety of recent work on power conservation in video encoding has focused on low-level device and circuit design. However, it has been generally recognized that high-level management of algorithms and architectures plays the most important role in power management of a mobile device.

2.1.2.1 SIMD Utilization

Most video coding algorithms operate on data that is highly capable of parallelism (such as motion estimation, motion compensation, and deblock filtering). Using the architectural characteristics of modern processors, the data level parallelism can be exploited by introducing additional logic to partition a higher precision data path to multiple pieces of packed, lower precision data and handling the pieces with a single instruction. SIMD (Single Instruction Multiple Data) is such a technique in which one instruction performs the same operation on multiple data elements, in parallel. Typical examples of parallel data processing with multimedia support are Intel's MMX (Multimedia Extension to the Intel architecture), and Sun Microsystem's VIS (Visual Instruction Extension to the sun UltraSPARC architecture). Both MMX and VIS exploit the parallelism inherent in many multimedia algorithms and can enhance the computational performance of the processor [4][47][48][90]. By using one single instruction to process the data in parallel, a significant number of instruction is saved, which results in a big power saving. However, the speedup of using SIMD instruction

set is usually restricted by the overhead operations including data alignment, data copying and data type convention.

2.1.2.2 Algorithm Design

When it comes to complexity reduction, it is always more desirable to develop a fast algorithm. For example, other types of optimizations probably cannot make linear sort (an $O(N^2)$ algorithm) compete with heapsort (O(Nlog(N))) for long sequences. Thus, algorithm design and optimization for video coding has been an import research topic for years, especially for video applications with limited CPU processing power. Since motion estimation and DCT/IDCT are the two most computationally intensive aspects of video coding, the design of fast motion estimation and DCT/IDCT algorithms abstains the main attention.

Motion estimation (ME) is the process of finding motion vectors during the encoding process. Motion vectors provide displacements into the past and/or future reference frames containing previously decoded pixels that are used to form the prediction and the error difference signal. It is well known that the full search ME approach is computationally intensive, and therefore the search for an effective ME algorithm to reduce the computational complexity has been a challenging problem. A great amount of fast ME algorithms have been proposed in the literature [37]. As a summary, fast ME algorithms can be classified as shown in Figure 2.1 [61].

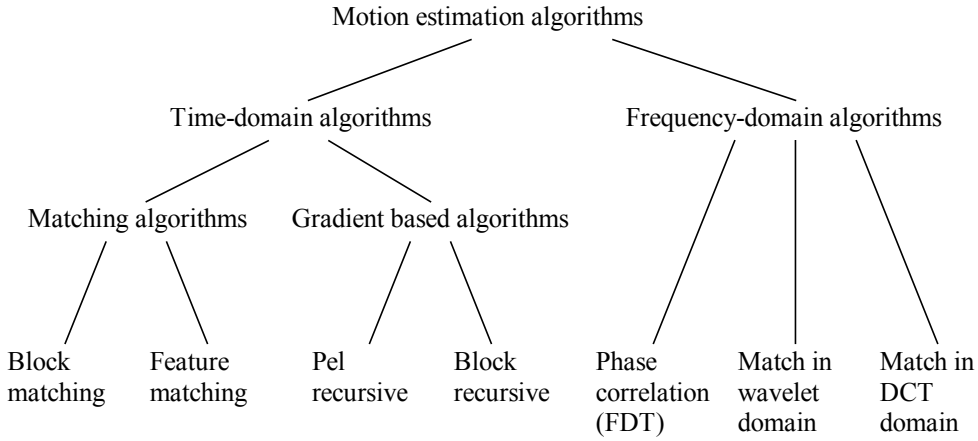


Figure 2.1 Fast ME algorithms Classification.

The discrete cosine transform (DCT) is a real-value frequency transform similar to the discrete Fourier transform (DFT). 2D DCT/IDCT has been widely used in video coding standards to remove the spatial redundancy of the video signal. The 2D DCT/IDCT can be computed using the row-column decomposition method by applying the 1D DCT/IDCT by rows and, then, by columns [71], [13], or directly computed without decomposing it into two successive 1D DCTs [24], [31], [36]. An alternative technique is based on the use of systolic architectures [19], [64]. In this case, the transformation is obtained by applying the 1D DCT twice. The book [77] provides an extensive introduction and an in-depth analysis of the properties, the various algorithms and the applications of the DCT.

2.2 Power-Distortion Optimized Video Coding

2.2.1 Design of Complexity Scalable Video Coding

To allow flexible control on the video encoder, complexity scalability must be introduced. In complexity scalable video coding, the computational complexity of the

video encoder can be adjusted by changing its complexity parameters. The design of a complexity scalable video encoder includes software implementations and hardware implementations.

In [21], a hierarchical block motion estimation algorithm based on partial distortion measure is proposed. It uses a coarse to fine approach to refine the search for each higher level, by dividing the motion vector search into several levels in a way that lower levels use partial distortions with higher decimation ratios. This algorithm can provide different computational complexities, i.e., different levels of power consumption, with different motion estimation precisions. In [49], a flexible framework is presented for DCT-based video encoding. In this framework, each of the encoding components (DCT and ME) features a set of parameters that can be used to control the computational complexity and performance and allow the encoder to run in real-time on machines with different computing power levels. [12] proposes a partially predefined configuration architecture for DCT and ME. [16] presents a low power video encoder with power, memory and bandwidth scalability for use in portable video applications. Scalable compression is achieved by using block transform based vector quantizer encoders implemented with table lookups. It can change its power consumption depending on the available channel bandwidth and can also trade-off bandwidth for power. In [20], the authors emphasize on optimizing the power consumption of the video coding co-processor design by minimizing computational units along the data path. Recently, in [3], depending on the battery power level, a number of B, P, I frames are discarded to reduce the transmission bits. The basic idea relies on conserving energy

by reducing the number of transmitted bits in a video stream. [63] utilizes a multi-stage coded modulation to accommodate rates in different modes. The authors state that, by judiciously selecting operating modes in response to mobile environment, lower energy consumption can be achieved.

However, the complexity parameters in the reviewed methods are obtained empirically and are lack of adaptability to the coding environment. Moreover, the techniques are designed to meet specific requirements for specific applications and video coding standards. To achieve the best performance, these parameters need to be re-obtained for the specific encoder used in the applications. On the other hand, the reviewed encoders can only provide complexity scalability to some extents, which limits its applications. Until now, no general complexity scalable video-coding framework has been proposed.

2.2.2 Modeling of Power Consumption and Rate-Distortion

To better understand the power consumption behavior of a video encoder, a power-consumption model is desirable. The modeling provides an estimation of the actual consumed power, which helps system analysis and system design. A few research works are reported in this area. [62] describes a novel approach for power-aware scheduling of multimedia tasks in RTOSs (Real Time Operating System) that trades off computation deadlines against power consumption. The proposed approach uses history-based prediction of multimedia sample processing time to aggressively adapt the processor voltage and frequency for significant power consumption reduction. In [17], a set of experiments is performed to understand the energy usage pattern of handheld

devices while decoding and displaying MPEG compressed video in software. Experiments are designed so as to bring forth parameters that can be used to predict the energy requirement for MPEG playback. In [69], a linear model is used to model the power consumption required for computing motion estimation, DCT, and quantization, of a H.263 video encoder, running with different motion estimation algorithms. The power consumption model is validated by measurement data.

For video coding under bit rate constraints, the goal is to maximize the picture quality, i.e., minimize the coding distortion, for a given bitrate budget. Thus, it is necessary to model the rate-distortion (R-D) relationship. R-D modeling based on statistical properties of the source data [41], [43], or empirical analysis on the observed data [66], [89] has been an active research activity in video coding and communication for the past few decades, and results in various bit rate control algorithms, such as [22], [15], [29]. However, only the relationship of rate and distortion was studied. In the increasing number of new pervasive computing applications, the complexity of the codec needs to be taken into account since these applications usually have strict complexity requirements. It is therefore desirable to develop appropriate models to study and understand the interaction and tradeoffs between rate, complexity, and distortion.

2.2.3 Video Coding Optimization

Depending on the application and its requirements, different constrained optimization problems have been formulated and analyzed. The optimization goal can

be the minimal distortion constrained by power supply and minimal power consumption constrained by expected distortion.

2.2.3.1 Distortion Optimized Video Coding

The problem in this category is to minimize the distortion or maximize the quality while the power supply is limited. The goal in [83] is to minimize the amount of distortion in the reconstructed video sequence under certain channel bandwidth and transmission power constraints, with transmission power allocated across packets. On the other hand, [84] considers the optimum allocation of transmission rate, source coding rate, and channel code rate under a given power constraint. In [18], a joint source coding and power control approach to simultaneously maximize the per-cell capacity while maximizing the quality of the delivered video to individual users subject to a constraint on the total available bandwidth is proposed. An interesting work is reported in [60], where a power-distortion optimized coding mode selection scheme is proposed for variable bit rate videos in wireless code-division multiple-access (CDMA) systems. The author proposes an optimum mode-selection algorithm for transmitting digital video over a CDMA channel, which minimizes the distortion for a given power consumption value. Unlike conventional R-D optimized mode-selection methods in which the Lagrangian multiplier depends on the channel conditions, the proposed scheme takes into account time-varying channels.

2.2.3.2 Power Optimized Video Coding

Currently, there is an active field of research that focuses on minimizing transmission energy/power under quality of service requirements for wireless video

communication, which is generally formulated as a constrained optimization problem to minimize the total end-to-end consumed power constrained by the video distortion and transmission bandwidth requirements.

In [86], a joint source coding and transmission power allocation scheme is proposed. In this scheme, no channel coding is used, and the transmission power is allocated adaptively to different video segments based on their relative importance. The optimization problem is formulated such that the transmission energy required to transmit a video frame is minimized at some acceptable average decoded video quality. Alternatively, [33] formulates an optimization problem that corresponds to minimizing the energy required to transmit a video frame with an acceptable level of distortion. In [78], the transmitter wants to obtain a joint power and coding-rate selection in order to maximize a video quality metrics within an analytical model. An optimum power-management scheme has been proposed, based on the model of the end-to-end distortion of INTRA frame refreshed image sequences. In [88] and [58], the authors consider the processing power for source coding and channel coding as well as transmission power for a given video quality and propose a power-minimized bit-allocation scheme. By changing the accuracy of motion estimation, different power and distortion levels for a H.263 encoder are provided. The search parameter of ME, source rate, channel rate, and transmission bit energy are obtained by solving the optimization minimization problem to achieve the minimal power consumption for a given expected end-to-end distortion and available total rate. This work is extended in [87], considering the interference to other users when performing the optimization of power consumption.

In [59], the proposed scheme minimizes the transmit power subject to the distortion constraint resulted from the loss of each packet and the error propagation effect caused by the motion compensation. It implicitly controls the target bit-error rate (BER) of the video packet according to the importance of the packet and the channel-induced distortion. However, the processing power is not considered in this paper. In [68], the authors introduce an approach for minimizing the total power consumption of a mobile transmitter due to source compression, channel coding and transmission subject to a fixed end-to-end source distortion. Unlike other approaches using the distortion determined from actually coded and transmitted video sequences (albeit the transmission process is simulated based on some models), two parameters, the quantization step size and the INTRA update rate, are used to control the source coder bit rate and the power consumption. The end-to-end distortion is determined based on an analytical model reported in [82].

CHAPTER 3

COMPLEXITY SCALABLE VIDEO CODING DESIGN

3.1 Motivation

As the first step of power and distortion optimized video coding, the encoder needs to be parameterized to enable flexible control on power consumption and video distortion. In other words, we need to introduce some complexity parameters into the encoder to control the coding behavior of the major encoding modules. Thus design of the complexity scalable video coding structure is one of the most important criteria for power consumption and video distortion control for pervasive computing applications.

3.2 Video Coding Complexity Analysis

3.2.1 Video Encoder Structure Overview

Typical video encoders, including all the standard video encoding systems, such as MPEG-2, H.263, and MPEG-4, employ a hybrid motion compensated DCT encoding scheme, which is summarized as the following. A video frame is divided into macroblocks (MBs). Each MB consists of four blocks (8×8 pixels) of luminance and one block for each of the two-chroma components. The processing of a video frame is done at the MB level. To exploit the temporal dependencies of MBs between successive frames, ME/MC is done through inter picture prediction. Transform coding of the residual prediction error signal, such as the DCT, is used to exploit the spatial

redundancy. After DCT is performed, the coefficients are numbered in a zig-zag order from the top left to bottom right. Scalar quantization is then applied to the resultant DCT coefficient matrix. Quantization is the lossy component of video compression. It simply reduces the number of bits needed to store the transformed coefficients by reducing the precision of those values. The resulting data are entropy encoded using a Huffman variable word length scheme in a lossless manner to give better overall compression gain. To decompress the image, the process is carried out in reverse. Among all the coding operations of video coding, the ME and DCT has been shown to have high complexity.

DCT/IDCT is typically done on each 8x8 block (Note that in the new H.264/AVC, DCT is done on 4x4 blocks). 1-D DCT requires 64 multiplications and for an 8x8 block 8 1-D DCTs are needed. 2-D DCT requires 54 multiplications and 468 additions and shifts. IDCT has similar complexity as DCT.

The objective of ME is to find the best match of a reference block in the reference frame that yields the minimum block distortion measure (BDM) within a given search window. Because of its simplicity, the sum of absolute difference (SAD) between the reference block and current block is popularly used as the BDM. For the SAD-optimal ME problem, the search for the optimal motion vector (MV) can be expressed as:

$$(u_0, v_0) = \text{argmin } SAD(u, v), \tag{3.1}$$

where (u_0,v_0) is the optimal MV, representing the horizontal and vertical displacement respectively, and *SAD(u, v)* is the SAD value of the candidate motion vector *(u, v)* within the search window, given by:

$$SAD(u,v) = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left| f_{t-r}(i+u, j+v) - f_t(i,j) \right|, \tag{3.2}$$

where $f_{t-r}(\cdot,\cdot)$ and $f_t(\cdot,\cdot)$ refer to the blocks with size of $M \times N$ in the reference frame and current frame respectively. One SAD computation requires $M \times N$ subtractions, $M \times N$ absolute value operations, and ($M \times N$ - 1) additions. As 3.1 shows, the MV has the minimum distortion, chosen from a certain number of MV candidates. In full search ME, all the candidates within the search window need to be evaluated. Remember that evaluating each MV candidate in the search window requires one SAD computation, which makes the MV search quite computationally complex.

Figure 3.1 illustrates the structure of a typical hybrid block based video encoder. There are several major encoding modules: motion estimation (ME) and compensation (COMP), DCT, quantization (QUANT), entropy encoding (ENC) of the quantized DCT coefficients, inverse quantization (DQUANT), inverse DCT (IDCT), picture reconstruction (RECON), and interpolation (INTERP) [81]. For the ease of exposition, the DCT, IDCT, QUANT, DQUANT and RECON modules are collectively referred to as PRECODING, which can be considered as the data representation module. In this way, the video encoder has only three major modules: ME, PRECODING, and ENC.

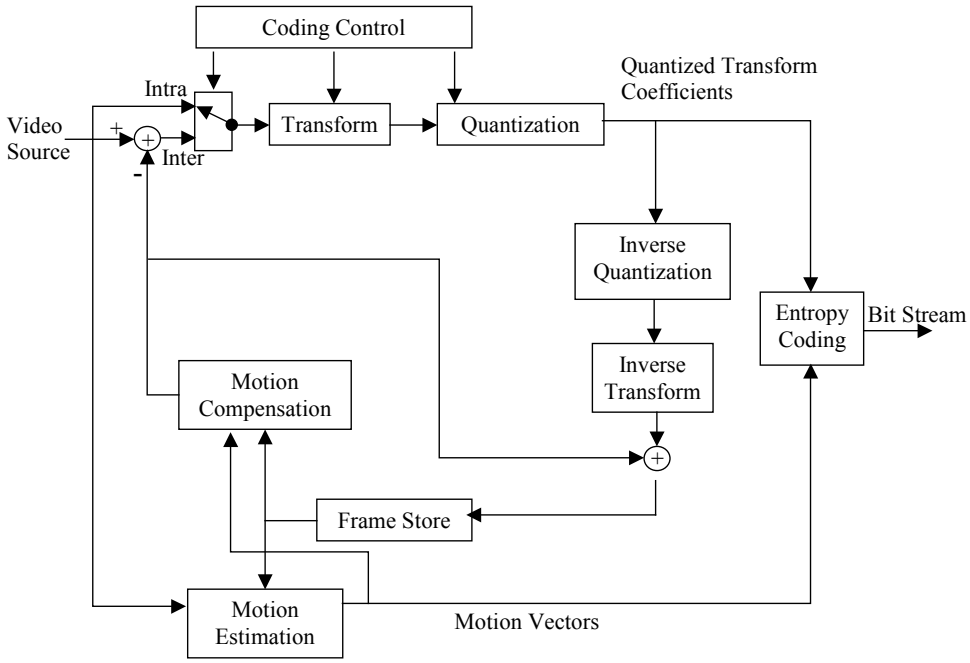


Figure 3.1 Video Coding Structure.

3.2.2 Complexity Profile

In order to find out the most computational expensive component of a video encoder and obtain the maximum gain, it is necessary to perform "profiling", i.e. analyzing the complexity distribution in different parts of the system. To analyze the run-time complexity of the major encoding modules, we run a MPEG-4 video encoder using full search ME on a Dell dimension 8200 PC, equipped with Intel Pentium 4@ 2GHz, 1GB and profile its computational complexity, measured as the average processor cycles. The test video sequences are "Akiyo", "Stefan", "News", and "Bream". In Table 3.1 and Table 3.2, we list the percentage CPU occupancy of the major encoding modules for Simple Profile @ Level 1 and Core Profile @ Level 2 respectively. Figure 3.2 and Figure 3.3 show the corresponding pie charts. We also run

the experiments on the new H.264/AVC encoder with QCIF format sequences encoded at 15 fps and QP = 27 using 1 reference frame and 5 reference frames respectively. The results are shown in Table 3.3, Table 3.4, Figure 3.4, and Figure 3.5. We have also evaluated the encoder CPU occupancy with other video sequences and different frame rate and bit rate settings. Only a slight difference has been observed.

Table 3.1 CPU occupancy of the major encoding components for a MPEG-4 encoder, Simple Profile @ level 1

Sequences	ME exclude SAD (%)	SAD (%)	FDCT (%)	IDCT (%)	TEXTURE CODING (%t)	QUANT/ DEQUANT (%)	MISCELLANOUS (%)	Total Time (ms)
Akiyo	26.3	64.6	1.6	1.6	0.5	0.4	5.0	90944.4
Stefan	13.3	82.3	1.0	1.0	1.0	0.2	1.2	289878.6
Average	16.4	78.1	1.1	1.1	0.9	0.2	2.1	

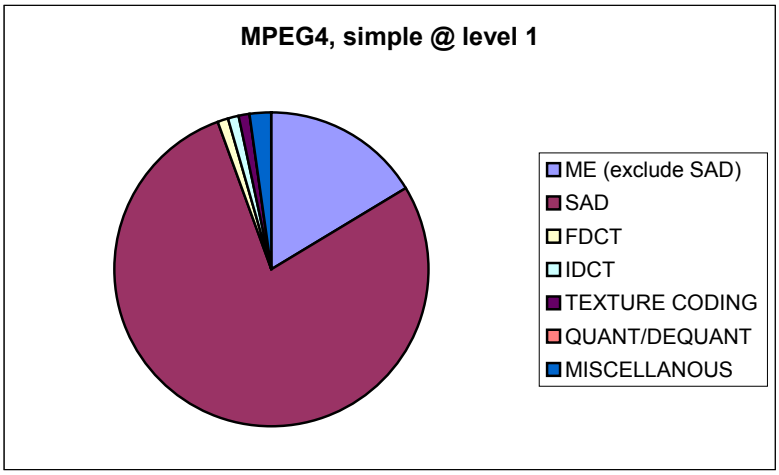


Figure 3.2 Complexity distribution of a MPEG-4 encoder, Simple@level1.

Table 3.2 CPU occupancy of the major encoding components for a MPEG-4 encoder, Core Profile @ level 2

Sequences	ME exclude SAD (%)	SAD (%)	FDCT (%)	IDCT (%)	TEXTURE CODING (%t)	QUANT/ DEQUANT (%)	MISCELLANOUS (%)	Total Time (ms)
News	25	28.1	2.6	2.7	11.1	0.5	30.0	344952.5
Bream	34.3	25	1.9	1.9	8.3	0.4	28.2	14182.6
Average	25.4	28.0	2.6	2.7	11.0	0.5	29.9	

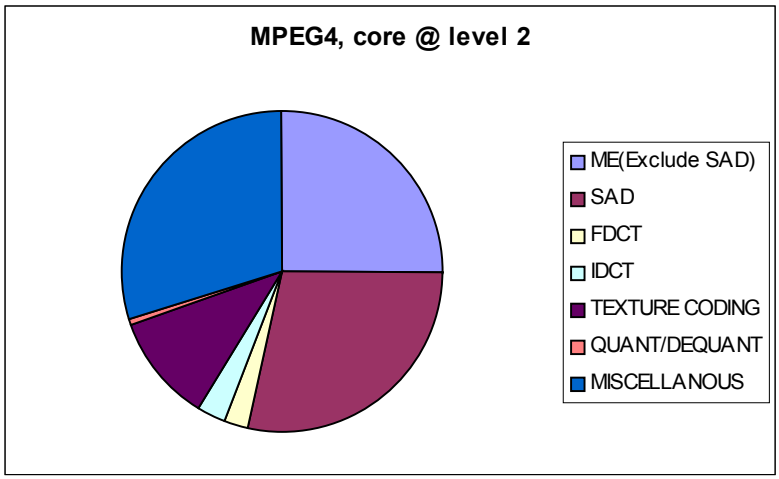


Figure 3.3 Complexity distribution of a MPEG-4 encoder, Core@Level2.

Table 3.3 CPU occupancy of the major components for a H.264 encoder, ref=1

Sequence (Ref =1)	ME(exclude SAD)	SAD	Mode Decision	DCT/IDCT, QUANT/DEQUANT	Entropy Coding (CABAC)	MISCELLANOUS
Akiyo	47.80	37.20	2.70	3.10	4.60	4.60
Stefan	47.20	33.80	2.60	3.40	8.50	4.50
Average	47.50	35.50	2.65	3.25	6.55	4.55

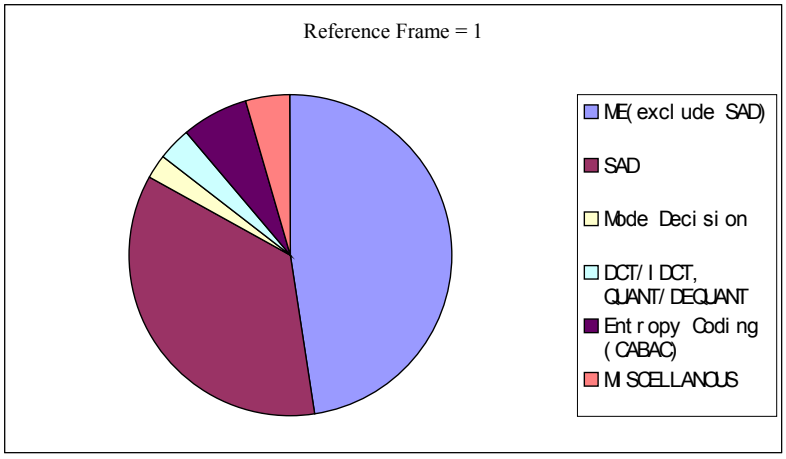


Figure 3.4 Complexity distribution of a H264 encoder, ref=1.

Table 3.4 CPU occupancy of the major components for a H.264 encoder, ref=5

Sequence (Ref =5)	ME(exclude SAD)	SAD	Mode Decision	DCT/IDCT, QUANT/DEQUANT	Entropy Coding (CABAC)	MISCELLANOUS
Akiyo	49.30	33.80	2.00	2.60	3.60	8.70
Stefan	47.10	37.80	2.10	2.80	6.60	3.60
Average	48.20	35.80	2.05	2.70	5.10	6.15

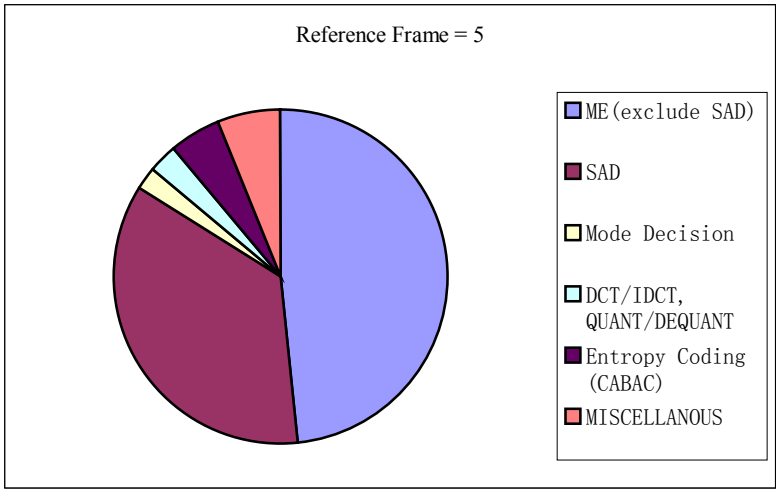


Figure 3.5 Complexity distribution of a H264 encoder, ref=5.

It can be seen that the ME process, which includes the SAD computation, is the most computation-intensive module, consuming most of the processor cycles. Following ME, the PRECODING modules collectively take the second place of the total processor cycles consumption. The ENC module, which is basically a bit splicing engine, uses a relative small amount of the total CPU time. In addition, its computational complexity mainly depends on the coding bit rate. Moreover, since the output bitstream must be conformed to bitstream syntax defined by the specific video standard, parameterized coding is not performed in the component of entropy coding. Thus, to gain the maximum flexibility in complexity control, we introduce parameterized video coding into the ME module and the PRECODING module.

3.3 A Complexity Scalable Video Encoder

3.3.1 Complexity Scalable ME Design

From 3.1, we can see that the ME process is simply a sequence of SAD computations to find the MV that has the minimum SAD. Note that the computational complexity of each SAD computation is a constant. Therefore, the overall computational complexity of the ME module is linearly proportional to the number of SAD computations, denoted by λ_{ME}. In the proposed energy scalable framework, λ_{ME} is determined by system-level power management and quality optimization. Since ME is performed on the MB level, therefore at the frame-level we need to allocate the current available λ_{ME} SAD computations among the MBs in the video frame.

3.3.1.1 Dynamic Allocation of the SAD Computations

It is well known that the moving objects in the video scene contribute most to the overall visual quality. This suggests that for ME with limited number of SAAD computations, we need to allocate the available λ_{ME} SAD computations among the MBs according to their motion characteristics to optimize the overall picture quality. In this work, we use a *motion history image (MHI)* to allocate the SAD computations among the MBs.

The *MHI* captures the history of motion activity. Each pixel *(i,j)* in an *MHI* corresponds to the spatial $(i,j)^{th}$ block in a sequence. The pixel intensity describes how long there has been no motion detected at that location since the last observed motion. Let $I_k(i,j)$ be the pixel intensity at time index *k*. It is initially set to zero, i.e., $I_0(i,j) = 0$. At each frame, the pixel intensity is updated by a simple rule, given by:

$$I_k(i,j) = \begin{cases} I_{k-1}(i,j)+1 & |mv_x^k(i,j)|+|mv_y^k(i,j)|=0 \\ 0 & |mv_x^k(i,j)|+|mv_y^k(i,j)|\neq 0 \end{cases}, \tag{3.3}$$

where $(mv_x^k(i,j), mv_y^k(i,j))$ is the MV of block *(i,j)* at frame *k*. As expressed by (3.3), the *MHI* can be considered to be a histogram of the areas of non-moving regions. For ease of expression, we omit *k* in the following.

Figure 3.6 illustrates one example of the *MHI* of video sequence "Akiyo". For display purpose, each pixel of the motion history image is depicted as one block. The pixel intensity is normalized to the range of 0 to 255. Note that the brightness represents how long no motion activity is detected since last witnessed motion. From Figure 3.6, we can observe that the motion history image not only contains motion history

information but also gives a good estimation of the spatial distribution of motion activity. The spatial distribution of motion activity indicates the location of "active" regions and static regions. Once these regions are identified, refined fast ME techniques can be applied to different regions in order to obtain better MVs with faster speed. Maintaining a *MHI* incurs low overhead because it can be obtained by using the MVs already determined after ME. Compared with some of the traditional region detection techniques [23], [73], no extra computation or expensive operations are required.

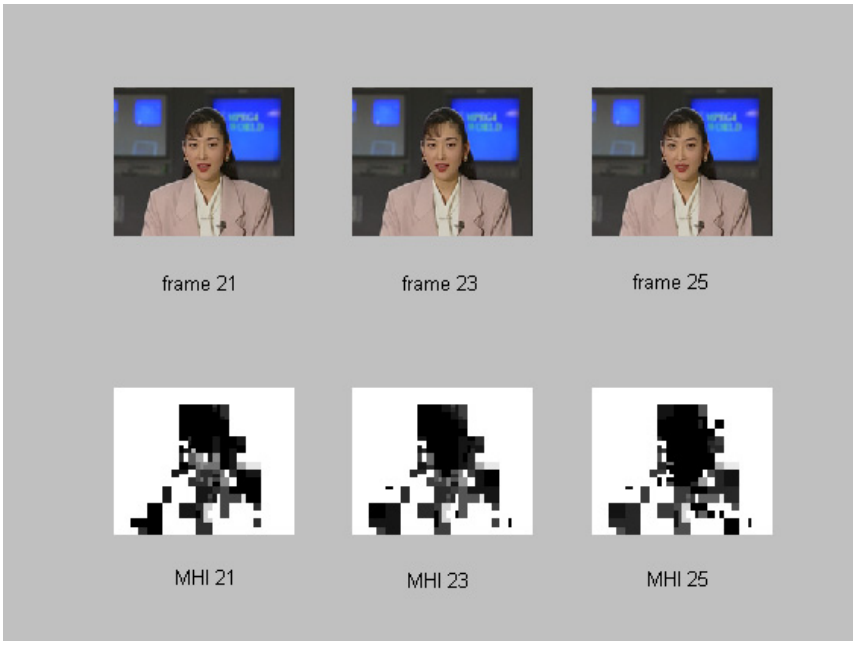


Figure 3.6 Motion history image of sequence "Akiyo".

Let $M=[m_{ij}]_{MR*MC}$ denote the *MHI*, where *MR* and *MC* are the numbers of MB's per row and per column, respectively. m_{ij} is the pixel intensity of *MHI* for the $(i,j)^{th}$ MB in current frame. The larger the value of m_{ij}, it is of higher probability that this MB is a static block, and less SAD computations can be allocated to this MB. The number of SAD computations allocated to the $(i,j)^{th}$ MB, denoted by $nsad_{ij}$, is determined by

$$nsad_{ij} = \frac{1}{N-1}\left[1 - \frac{m_{ij}}{\sum_{(k,l)\geq(i,j)} m_{kl}}\right] \cdot N_{sad} \quad , \tag{3.4}$$

where N is the number of MBs left so far that need to perform the motion estimation, and N_{SAD} is the available number of SAD computations. Here, *N-1* is a normalization factor, because

$$\sum_{(i,j)}\left[1 - -\frac{m_{ij}}{\sum_{(k,l)\geq(i,j)} m_{kl}}\right] = N-1, \tag{3.5}$$

Initially, *Nsad* is set to be λ_{ME}. Suppose the motion search range is SR. If $nsad_{ij}$ >= $(2*SR+1)^2$; it means the computational power is enough to perform a full search for this block. Otherwise, the complexity controllable ME scheme described in the following is used to find the MV, whose complexity is controlled by $nsad_{ij}$.

3.3.1.2 Complexity Controllable Motion Estimation Scheme

The statistical behavior of the SAD error surface has a significant impact on the performance of the fast search algorithm for block matching motion estimation. Most conventional fast algorithms have explicitly or implicitly assumed that the error surface is unimodal over the search window. Unfortunately, this assumption is not always true in practical applications and the MV search is likely to get trapped into a local minimum. Thus, when the computation power is available, more search positions need to be checked to verify the MV in order to converge to the global minimum. In contrast, when the computation power is low, if the potential improvement in video quality by

searching more positions is not computationally justified then the search can be stopped earlier.

The proposed adaptive search scheme is based on the popular media-bias diamond search wherein 4 points around the current minimum are searched to find the next minimum in every iteration (Figure 3.7 (a)). For a given number of pre-allocated SAD calculations $nsad_j$, we denote SI_j the number of search iterations during the MV search, i.e., the number of diamond search patterns we need to perform. It is approximated by a linear relationship, given by:

$$SI_j = \psi * nsad_j, \tag{3.6}$$

The parameter ψ is adaptively updated for each block, since the motion characteristics and the SAD error surface are correlated to those of the adjacent blocks. Now, the proposed search strategy works as following. After the allocation of SAD computation, if $nsad_j$ is 0, which means we have no available computation power at all, the MV of the collocated MB in the previous coded frame is chosen as the MV for current block because of the high temporal correlation between current frame and the previous encoded frame. If $nsad_j$ equals to 1, the MV is chosen from the median MV predictor and the temporal previous MV that yields the smaller SAD value. In other cases, the number of search iteration is determined by Equation (3.6) and a parameterized search is used to find the motion vector, summarized as following:

- *Step 1*: We start with the diamond search pattern with size 1, showed at Figure 3.7 (a), and continue the search until the search centre is found with the

minimum SAD. If the number of search iteration exceeds SI_j, we stop the search.

- *Step 2*: We switch to the diagonal search with the size of the current diamond size, showed at Figure 3.7 (b), and continue the search. If the search centre is found with minimum SAD, we increase the diamond size by 1. Otherwise the diamond size is reset to 1. Go back to Step 1. During the search, whenever the number of search iteration exceeds SI_j, we stop the search.

- *Step 3*: The motion vector is returned and the parameter ψ is updated.

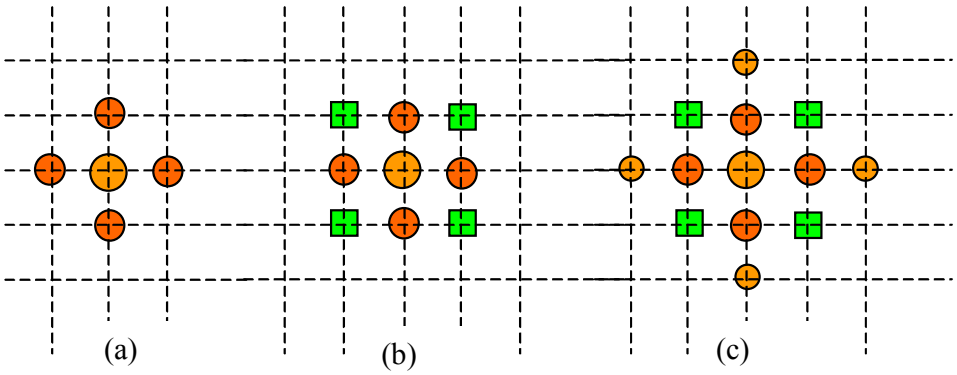


Figure 3.7 The search patterns. The biggest point has the minimum distortion. (a) The diamond search with size equal to 1. (b) The diagonal search with size 1, the square points are checked. (c) The diamond size is changed to 2. The outermost round points are checked.

Note that like other conventional fast ME algorithms, the partial distortion computation technique is also applied. In addition, the checked points during the search procedure are tracked to avoid the unnecessary checking when the search pattern moves.

3.3.2 Complexity Scalable PRECODING Design

In this section, we present a parametric complexity scalability scheme to collectively control the computational complexity of the PRECODING modules, namely, the DCT, QUANT, DQUANT, IDCT, and RECON modules.

In typical video encoding as illustrated in Figure 3.1, DCT is applied to the difference MB after motion estimation and compensation, or the original MB if its coding mode is INTRA. After the DCT coefficients are quantized, DQUANT, IDCT, and RECON are performed to reconstruct the MB for motion prediction of the next frame. In transform coding of videos, especially at low coding bit rates, the DCT coefficients in the MB might become all zeros after quantization. We refer to this MB as an all-zero MB (AZMB). Otherwise, it is called a non-zero MB (NZMB). In international standards for video encoding, such as MPEG-2, H.263, and MPEG-4, "non-zeros" also means the CBP (coded block pattern) value of the MB is non-zero. If we can predict an MB to be AZMB, all the above PRECODING operations can be skipped, because the output of DQUANT and IDCT of an AZMB is still an AZMB, and the reconstructed MB is exactly the reference MB used in motion estimation and compensation. Therefore, the encoder can simply copy over the reference MB to reconstruct the current MB. This is a unique property of the AZMB, which can be used to reduce the computational complexity of the video encoder.

In this work, the unique property of the AZMB is used to design a complexity scalability scheme for the PRECODING modules. Let $\{x_{nk} \mid 0<=n,\ k<=7\}$ be the coefficients in the different MB after motion estimation. For INTRA MB's, $\{x_{nk}\}$ are the

original pixels in the video frame. Let $\{y_{ij} \mid 0<=n, k<=7\}$ be the DCT coefficients. According to the definition of DCT, we have

$$y_{ij} = \frac{1}{4} C_i C_j \sum_{n=0}^{7} \sum_{k=0}^{7} x_{nk} \cos(i\pi \frac{2n+1}{16}) \cos(j\pi \frac{2k+1}{16}), \tag{3.7}$$

where

$$C_i = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } i = 0 \\ 1 & else \end{cases}, \; C_j = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } j = 0 \\ 1 & else \end{cases}. \tag{3.8}$$

We can see that

$$\left| y_{ij} \right|^2 \le \sum_{n=0}^{7} \sum_{k=0}^{7} \left| x_{nk} \right|^2, \tag{3.9}$$

Note that the right-hand side is the SAD of the difference MB, which is already computed during the motion estimation. This suggests us that the SSD could be an efficient and low-cost measure to predict the AZMB. After motion estimation and compensation, let $\{SSD_i \mid 1<= i <= M\}$ be the SSD values of the M MBs in the video frame sorted in an ascending order. In the proposed complexity scalability scheme for PRECODING, We force the first M- λ_{PRE} MBs to be AZMBs, and treat the remaining λ_{PRE} MBs as NZMBs to which the PRECODING operations are applied. Let C_{NZMB} be the number of processor cycles needed by the PRECODING operations to finish one NZMB. The value of C_{NZMB} can be obtained either by theoretical cycle estimation of the PRECODING modules, or from simulation statistics. In practice, the value of C_{NZMB} may vary slightly from MB to MB. Note that the power management and energy consumption control operate on a level much higher than the MB. For example, in real

world applications, it is sufficient to adjust the system power control parameters for every 5 seconds, which have 150 frames (if coded 30 frame per second) and thousands of MB's. At this level, in its average sense, it is quite reasonable to assume C_{NZMB} is a constant. The overall complexity of the PRECODING modules, denoted by C_{PRE} is then given by:

$$C_{PRE} = \lambda_{PRE} * C_{NZMB,,} \tag{3.10}$$

We refer to this type of complexity scalability scheme as λ_{PRE} scalability. As far as the subjective video quality is concerned, the proposed λ_{PRE} scalability performs reasonably well. As mentioned in previous Section 3.3.1.1, the moving objects in the scene contribute most to the video presentation quality, and have unique significance in subjective video quality evaluation. In motion estimation and compensation, these regions of the picture often correspond to blocks with relatively large SAD values. In the λ_{PRE} scalability and dynamic rate control scheme, the AZMB bits are added to these blocks, resulting in an improved visual quality in these regions.

Figure 3.8 shows the 100th frame of "Foreman" encoded at 192 kbps, and the 80-th frame of "Carphone" encoded at 64 kbps with 100% and 20% PRECODING complexity. Perceptually, we can hardly see much difference between them. It should also be noted that these blocks with SAD below the threshold often correspond to picture regions with smooth spatial or temporal variation. The slightly degraded quality in these regions can be easily restored by post-processing techniques, such as deblocking, deringing, or temporal smoothing, at the receiver side.

Figure 3.8 Coded video quality comparison for Frame 100 of "Foreman" and Frame 80 of "Carphone" when (A) Left: 100% blocks are encoded; (B) Right: 20% blocks are encoded.

3.4 Concluding Remarks

In this chapter, based on the complexity analysis of typical video encoding systems, we have developed a parametric video encoding architecture, which is fully scalable in power consumption. Unlike other research works, the proposed architecture can be applied to various video coding standards and used in different applications. Moreover, by using the dynamic SAD computation allocation and AZMBs allocation, we are able to fully control the coding complexity while maximizing the video quality.

CHAPTER 4

POWER-RATE-DISTORTION ANALYSIS

4.1 Motivation

Video encoding and data transmission are the two dominant power-consuming operations in wireless video communication, especially over wireless LAN, where the typical transmission distance ranges from 50m to 100m. Experimental studies show that for relative small picture sizes, such as QCIF (176x144) videos, video encoding consumes about 2/3 of the total power for video communication over wireless LAN [3][67]. For pictures of higher resolutions, it is expected that the fraction of power consumption by video encoding will become even higher. From the power consumption perspective, the effect of video encoding is two-fold. First, efficient video compression significantly reduces the amount of the video data to be transmitted, which in turn saves a significant amount of energy in data transmission. Second, more efficient video compression often requires higher computational complexity and larger power consumption in computing. These two conflicting effects imply that in practical system design there is always a tradeoff among the bandwidth R, power consumption P, and video quality D. Here, the video quality is often measured by the mean square error (MSE) between the encoded picture and original one, also known as the source coding

distortion. To find the best trade-off solution, we need to develop an analytic framework to model the power-rate-distortion (P-R-D) behavior of the video encoding system.

To our best knowledge, there has been no analytic framework for modeling the P-R-D behavior of the video encoding system. Rate-distortion (R-D) analysis has been one of the major research focus in information theory and communication for the past few decades, from the early Shannon's source coding theorem for asymptotic R-D analysis of generic information data [9], to recent R-D modeling of modern video encoding systems [22], [45], [46], [26]. For video encoding on the mobile devices and streaming over the wireless network, it is needed to consider another dimension, the power consumption, to establish a theoretical basis for R-D analysis under energy constraints. In energy-aware video encoding, the coding distortion is not only a function of the encoding bit rate as in the traditional R-D analysis, but also a function of the power consumption P. In other words,

$$D=D(R; P), \tag{4.1}$$

which describes the P-R-D behavior of the video encoding system. The P-R-D model provides a theoretical basis, as well as a practical guideline, for system design and performance optimization. Using the P-R-D model, we can perform energy consumption control on the underlying device at the system level. For example in a wireless sensor network, we can perform across-node energy optimization and network lifetime maximization.

In this chapter, we develop an analytic framework to model, control and optimize the P-R-D behavior of typical video encoding systems. We analyze the video

coding architecture presented in the previous chapter and perform R-D analysis of the ME and PRECODING modules. The integration of the R-D models for the complexity control parameters results in a comprehensive P-R-D model for the video coding system. Based on the P-R-D model, we develop a quality optimization scheme to determine the best configuration of complexity control parameters according to the power-supply level of the device to maximize the video presentation quality.

4.2 Power Consumption Analysis

As discussed in CHAPTER 3, we introduce several encoder parameters to control the computational complexity of the major encoding modules. Specifically, in this work, the complexity control parameter for the ME module is the number of SAD (sum of absolute difference) computations per frame, denoted by λ_{ME}. The computational complexity of ME, denoted by C_{ME}, is simply given by:

$$C_{ME} = \lambda_{ME} \times C_{SAD,} \tag{4.2}$$

where C_{SAD} represents the complexity of one SAD computation between the current MB and its reference MB. Here, the computational complexity is measured by the number of processor cycles used by the operation.

The computational complexity of all the PRECODING modules is controlled using one single parameter λ_{PRE}, which is the number of non-zero MB's in the video frame. Let C_{NZMB} and C_{PRE} be the PRECODING computational complexity of one non-zero MB (NZMB) and the whole video frame, respectively. From Section 3.3.2, we see that,

$$C_{PRE} = \lambda_{PRE} . C_{NZMB} \quad (4.3)$$

The ENC module, as a variable length-coding (VLC) engine, mainly consists of VLC table look-up and bit splicing of the codewords. The computational complexity of the ENC module, denoted by CENC, is approximately proportional to R. Therefore, we have:

$$C_{ENC} = S^* R * C_{BIT,} \quad (4.4)$$

where C_{BIT} is the per bit ENC complexity, and S is the size of the picture. Here, S is needed because R represents the coding bit rate in the unit of bits per pixel. The computational complexity of the video encoder, denoted by C and measured by the number of processor cycles per second, is given by:

$$C(R; \lambda_{ME} , \lambda_{PRE} , \lambda_F) = \lambda_F \times (C_{ME} + C_{PRE} + S \times R \times C_{BIT}) \quad (4.5)$$

where λ_F is the encoding frame rate. This model presents a complexity-scalable architecture for video encoding, whose computational complexity is mainly controlled by the parameter set $\{\lambda_{ME} , \lambda_{PRE} , \lambda_F\}$. It can be seen that, in the proposed complexity scalable video coding design, we try to find the "atom operations" that have fixed computational complexity, and decompose the overall video encoding into these atom operations. Specifically in this work the atom operations are the MB SAD computation, the PRECODING of one MB, and the per-bit ENC operation.

Let Φ be a mapping function to translate the computational complexity of the video coding system into corresponding power consumption. We can derive the relationship between the power consumption and the complexity control parameters,

$$P = \Phi(\lambda_F \times (C_{ME} + C_{PRE} + S \times R \times C_{BIT})). \quad (4.6)$$

4.3 R-D Analysis

4.3.1 ME Module R-D Analysis

To analyze the R-D behavior of the complexity control parameter λ_{ME}, we need to investigate the relation between λ_{ME} and the frame SAD S_f, which is the average SAD per pixel in the motion compensated difference frame. To this end, we collect the frame SAD statistics for different λ_{ME} from several test video sequences. Figure 4.1 plots the frame SAD S_f as a function of λ_{ME} for two QCIF video sequences: "Akiyo" and "Foreman". The simulation results suggest the following relation between λ_{ME} and S_f:

$$S_f(\lambda_{ME}) = \beta_0 + \beta_1 * e^{-\beta 2x}, \qquad x = \lambda_{ME} / \lambda^{max}_{ME}, \tag{4.7}$$

where the model parameters β_0, β_1, and β_2 are estimated by the statistics of previous frames; and $\lambda^{max}_{\mathrm{ME}}$ is the maximum value of λ_{ME}. Besides the SAD, another operation called SSD (sum of square difference), which is the square difference between the current MB and its reference, is often used in motion estimation. In hardware design, the SSD is more advantageous than the SAD because the subtraction and multiplication operations can be completed by a single instruction. In motion estimation, SAD and SSD have similar performance because SSD linearly increases with the SAD. Therefore, the proposed complexity control is also applicable to the SSD based ME. Simulation with SSD yields similar result as shown in Figure 4.1, and the complexity model in (4.7) also applies to SSD. In this case, the frame SSD S_f becomes the variance of the difference frame. From Section 4.4 we will see that the final P-R-D model needs the variance information for R-D analysis.

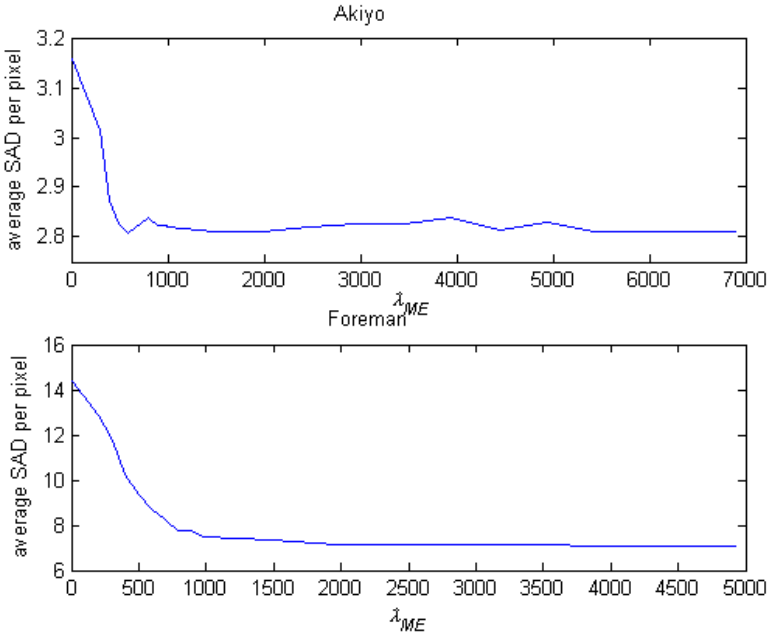


Figure 4.1 Frame SAD as a function of λ_{ME}.

4.3.2 PRECODING R-D Analysis

Using the mathematical framework for optimal bit allocation, we analyze the R-D behavior of the complexity control parameter λ_{PRE}, as discussed in Section 3.3.2. The dynamic rate control is a near-optimal bit allocation process. Let $\{\sigma_i^2 | 1 <= i <= M\}$ be the variance of the MB's in the video frame sorted in an ascending order. Let R be the target coding bit rate in bits per pixel (bpp). According to the classic R-D distortion formula, the distortion of the i^{th} MB is given by

$$D_i(R_i) = \sigma_i^2 \cdot 2^{-2\gamma R_i}, \tag{4.8}$$

where R_i is the bit rate of the i^{th} MB, and is a model constant. The optimal bit allocation can be then formulated as

$$D = \min_{\{R_i\}} \frac{1}{M} \sum_{i=1}^{M} \sigma_i^2 \cdot 2^{-2\gamma R_i} \tag{4.9}$$

$$\text{s.t.} \quad R = \frac{1}{M} \sum_{i=1}^{M} R_i . \tag{4.10}$$

The minimum distortion obtained by the optimal bit allocation is:

$$D = (\prod_{i=1}^{M} \sigma_i^2)^{\frac{1}{M}} \cdot 2^{-2\gamma R}, \tag{4.11}$$

In our complexity scalability scheme, the first M-λ_{PRE} MBs are encoded as AZMBs, while the remaining λ_{PRE} MB's are encoded as NZMBs. In this case, the bit rate of each AZMB is zero, and its coding distortion, denoted by D^z_i, is exactly the variance of the difference MB, i.e.,

$$D^z_i = \sigma^2_i * 2^{-2\gamma * 0} = \sigma^2_i, \quad 1 <= i <= M-L, \tag{4.12}$$

where $L=\lambda_{PRE}$ is introduced to simplify the notation. Since all the coding bits are allocated among the NZMBs, according to (4.12), the coding distorting of each NZMB, denoted by D^{nz}_i, is given by

$$D_i^Z = (\prod_{i=M-L+1}^{M} \sigma_i^2)^{\frac{1}{L}} 2^{-2\gamma \frac{MR}{L}}, M-L+1 \le i \le M \tag{4.13}$$

The overall distortion D of the video frame, which is average distortion of the AZMBs and NZMBs, is given by:

$$\begin{aligned} D = D(L) &= \frac{1}{M}[\sum_{i=1}^{M-L} D_i^2 + \sum_{i=M-L+1}^{M} D_i^{nz}] \\ &= \frac{1}{M}[\sum_{i=1}^{M-L} \sigma_i^2 + L(\prod_{i=M-L+1}^{M} \sigma_i^2)^{1/L} 2^{-2\gamma MR/L}]. \end{aligned} \tag{4.14}$$

To derive the expression for D(L), we consider the continuous-time version of (4.14). Note that $\{\sigma_i^2 | 1 <= i <= M\}$ is an increasing series. Figure 4.2 shows $\{\sigma_i^2\}$ for the 100th frame of the "Foreman". Experiments on other video frames and other video sequences yield similar results.

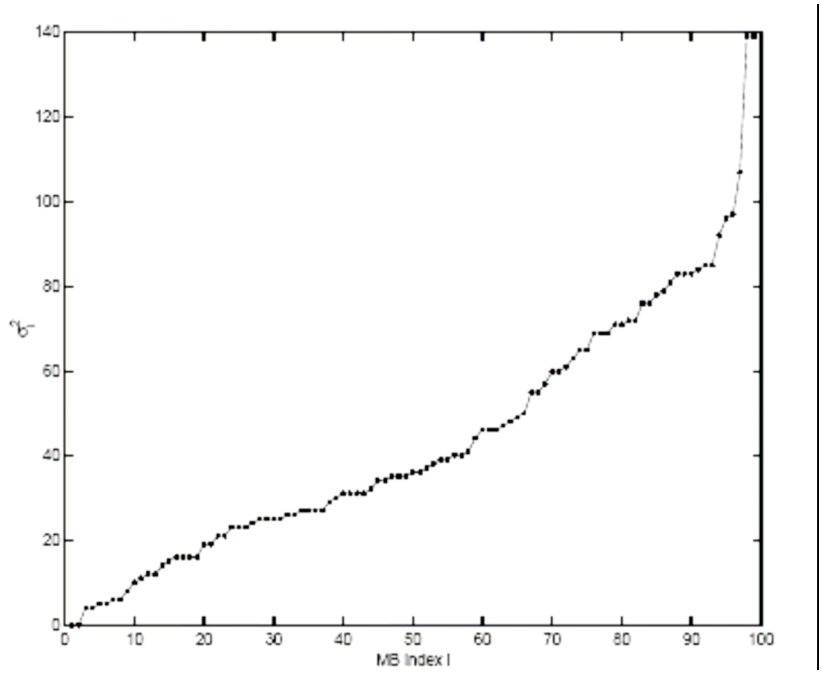


Figure 4.2 MB variances sorted in an ascending order for the 100th frame of "Foreman".

This suggests us that it is reasonable to model $\{\sigma_i^2\}$ with the following linear function

$$\xi(t) = A \cdot t, t \in [0,1], \tag{4.15}$$

such that

$$\sigma_i^2 = \xi(\frac{i}{M}), 1 \le i \le M \tag{4.16}$$

Here A is a positive constant. It should be noted that at the right end of the curve, the linear approximation is not accurate. However, since the R-D modeling is a statistical procedure to model the behavior of the whole frame, which has a large number of MBs, the approximation error within this small region will not affect much the performance of the whole model. Our simulation results that will be presented later confirm this observation. Similarly, we define *y=L/M*, and consider *D(y)* as the continuous-time version of {D(L)}, i.e.,

$$D(y)=D(L/M) \tag{4.17}$$

Note that the first term on the right-hand side of (4.21) can be written as:

$$\frac{1}{M}\sum_{i=1}^{M-L}\sigma_i^2 = \int_0^{1-y}\varsigma(t)dt = \int_0^{1-y} A\cdot t dt = \frac{A}{2}(1-y)^2, \tag{4.18}$$

where *y=L/M* represents the fraction of NZMBs in the video frame. Let $z=(\prod_{i=1}^{M}\sigma_i^2)^{\frac{1}{L}}$, we have:

$$\ln(Z)=\frac{M}{L}\cdot\frac{1}{M}\sum_{i=M-L+1}^{M}\sigma_i^2 = \frac{1}{y}\int_{1-y}^{1}\ln(At)dt = \ln A-\frac{1}{y}[y+(1-y)\ln(1-y)]. \tag{4.19}$$

Therefore,

$$D(y) = A(\frac{1}{2}(1-y)^2 + ye^{-\frac{1}{y}[y+(1-y)\ln(1-y)]}\cdot 2^{-2\gamma\frac{R}{y}} \tag{4.20}$$

4.4 The Power-Rate-Distortion Model

4.4.1 Parameters Estimation and Model Simplification

The R-D model for the PRECODING modules given by (4.20) has one parameter A. Note that:

$$\frac{1}{M}\sum_{i=1}^{M}\sigma_i^2 = \int_0^1 \varsigma(t)dt = \frac{A}{2}. \tag{4.21}$$

Therefore, A can be estimated by:

$$A = \frac{2}{M}\sum_{i=1}^{M}\sigma_i^2 = \frac{2}{M}\sum_{i=1}^{M} SSD_i \tag{4.22}$$

The R-D model in (4.20) will be used for energy consumption control and picture quality optimization. Since the model is highly nonlinear, it is not suitable for mathematical optimization. Therefore, we need to simplify the formulation , specifically the exponential term. Taylor expansion yields the following linear approximation,

$$e^{-\frac{1}{y}[y+(1-y)\ln(1-y)]} \approx (e^{-1}+e^{-3}) + (1-e^{-1}-e^{-3})(1-y). \tag{4.23}$$

Figure 4.3 shows the nonlinear exponential function (solid line) and its linear approximation (dashed line). It can be seen that approximation error is relatively small. With the linear approximation, the PRECODING C-R-D model becomes,

$$D(y) = A\cdot[\frac{1}{2}(1-y)^2 + y(1+a_0 y)\cdot 2^{-2\gamma\frac{R}{y}}], \tag{4.24}$$

where $a_0=e^{-1}+e^{-3}-1$.

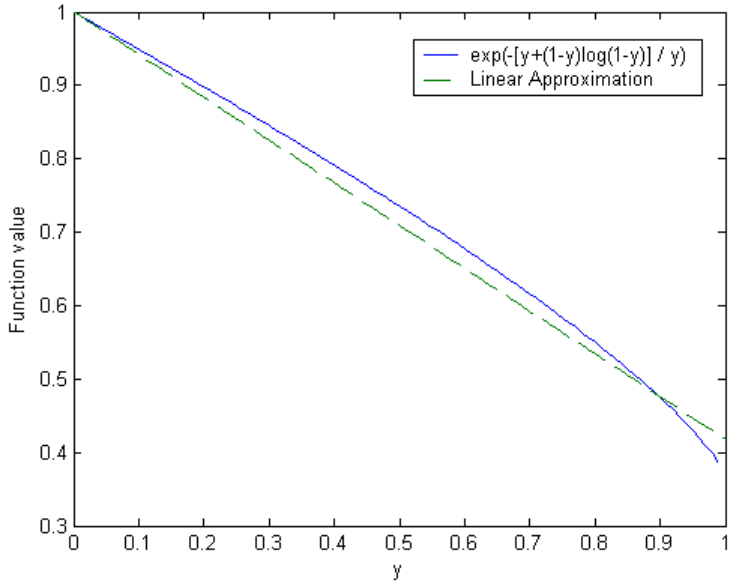


Figure 4.3 Linear approximation of $e^{-\frac{1}{y}[y+(1-y)\ln(1-y)]}$.

4.4.2 Integrated Power-Rate-Distortion Model

For a complexity target of λ_{ME} SSD computations, the average MB variance is given by

$$\frac{1}{M}\sum_{i=1}^{M}\sigma_i^2 = \beta_0 + \beta_1^{-\beta_2 x}, x = \frac{\lambda_{ME}}{\lambda_{ME}^{\max}}. \tag{4.25}$$

According to (4.29) and the PRECODING C-R-D model in (4.31), we have

$$D = D(R;x,y) = 2(\beta_0 + \beta_1^{-\beta_2 x})[\frac{1}{2}(1-y)^2 + y(1+a_0 y)\cdot 2^{-2\gamma\frac{R}{y}}], \tag{4.26}$$

where x and $y=\lambda_{PRE}/M$ are the normalized complexity control parameters. Both x and y range from 0 to 1, with 0 and 1 representing the lowest and highest computational complexity, respectively. It should be noted that the distortion in (4.26) only measures the quality for a single frame. The research in video quality evaluation suggests that the

video presentation quality should be measured not only by the spatial quality of a single frame, but also by the temporal quality in motion smoothness [6]. Therefore, the encoding frame rate λ_F plays a very important role in quality evaluation. It is also a key parameter in energy consumption control. For example, at lower frame rates, more energy can be allocated to each frame to improve the spatial quality. However, in this case, the temporal video quality degrades. Although many results have been published in subjective video quality evaluation, most of them focus on experimental studies. For quality optimization of video coding, we need an analytic, mathematically tractable model to describe the video presentation quality. The experimental results in [6] suggest that the video presentation quality D_v should consist of two parts: the spatial quality of a single picture D_{spatil} and the temporal motion quality $D_{temporal}$. D_{spatil} given by (4.26). $D_{temporal}$ depends on the encoding frame rate. In typical video decoding and display, if a video frame is skipped, the previous decoded picture stays on the screen until the next frame is decoded. In other words, the decoder reconstruction of the skipped frame is the copy of its previous decoded frame. From the video encoder point of view, the ME complexity x, the PRECODING complexity y, and the bit rate R of the skipped video frame are all zeros. Therefore, from (4.26), we can see that its coding distortion is given by

$$D_{temporal} = D(R; x, y)|_{R=0, x=0, y=0} = \beta_0 + \beta_1, \tag{4.27}$$

which is the MSE between the skipped frame and its previous reconstruction. Let ω_s and ω_t be the perceptual weight on the spatial quality and temporal quality, respectively. The experimental results in [6] suggest that ω_s and ω_t should be a function of the frame

rate. For example, if the video encoder encodes only one frame per minute, although each picture has very high quality, the viewer will complain about the bad video streaming service because he has missed a lot of important motion information and the spatial information in between. In this work, we choose the perceptual weight as follows,

$$\varpi_t=(1-z)^2;\ \varpi_s=1-\varpi_t; \tag{4.28}$$

where $z=\lambda_F/f_{max}$, and f_{max} is the maximum frame rate with a default value of 30 fps. Therefore, the video presentation quality is defined as

$$\begin{aligned} D_v &= \varpi_s D_{spatial} + \varpi_t D_{temporal} \\ &= (1-z)^2(\beta_0+\beta_1) + 2(2z-z^2)(\beta_0+\beta_1^{-\beta_2 x})[\frac{1}{2}(1-y)^2 + y(1+a_0 y)\cdot 2^{-2\gamma\frac{R}{y}}] \end{aligned}. \tag{4.29}$$

Let C_1, C_2, and C_3 be the constants in 4.6, we have the power consumption computed as:

$$P=\Phi(z(C_1x+C_2y+C_3R)), \tag{4.30}$$

For a given power supply level P and a given rate R, we need to find the best configuration of the complexity parameters for the ME and PRECODING modules to maximize the picture quality. Mathematically, this can be formulated as:

$$\min_{\{x,y,z\}} D_v(R;x,y,z),\ s.t.\ P=\phi(z(c_1x+c_2y+c_3R)). \tag{4.31}$$

The minimization parameters *(x, y, z)* can be obtained using binary search of the minimum point. Note that the battery often has an operational lifetime of several hours, several days, or even several weeks. Therefore, there is no need to adjust the power control parameters too often, say every second, because the power supply condition

doesn't change that quickly. Suppose the adjustment period is five seconds. This means we only need solve the R-D optimized power control problem in (4.31) once per five second. Therefore, the overhead of the power control is relatively small. In our future work, we shall investigate the possibility of further simplification of the model and its solution as well.

4.4.3 R-D Optimized Power-Scalable Video Encoding

Using the P-R-D model and the optimal configuration of the power control parameters, the video encoder is able to achieve the R-D optimized power consumption scalability. The R-D optimized power-scalable video encoder system operates as follows:

Step 1: Determining the model parameters: In (4.31), the ME model parameters β_0, β_1, β_2 are estimated from the statistics of previous frames using linear regression. a_0 is a constant determined by (4.26). The model parameter is also determined from the R-D statistics of the previous frames. At beginning stage, for example the first second of video encoding, no power control is applied, because the system has sufficient power supply.

Step 2: Optimization: Find the optimal complexity control parameters $\{x,y,z\}$ using (4.31). This step is executed only if the power control is triggered according to the adjustment frequency, for example, once per five seconds.

Step 3: Frame rate and ME complexity control: Set the encoding frame rate to be $\lambda_F = z \cdot f_{max}$. The available SSD computations for ME is given by $\lambda_{ME} = x \cdot \lambda_{ME}^{max}$. Using the MHI-based allocation scheme presented in Section 3.3.1.1 to allocate the SSD

computation among the MB's. Using the fast and efficient diamond ME scheme to find the motion vector and the minimum SSD for each MB. The number of diamond search layers is controlled by the allocated SSD computations.

Step 4: PRECODING complexity control: Find the $(1-y)\cdot M$ MB's with the smallest SSD values and forces them to be AZMBs. The PRECODING operation is applied to the remaining NZMBs. Dynamic rate control is used to reallocate the bits from the AZMBs to the NZMBs. It can be seen that the complexity of the major encoding modules is controlled by the parameter set $\{x,y,z\}$ to match the power supply level of the mobile device. At the same time, these parameters are configured according to the P-R-D model such that the overall video quality is optimized.

4.5 Experimental Results

To evaluate the performance of the P-R-D model and the power-scaling video encoding system, we implement the proposed P-R-D model and power scalability scheme in the public domain H.263+ encoder. Similar performance is expected for other coding systems, such as MPEG-2 and MPEG-4. In our simulations, the maximum search points for each MB λ_{ME}^{max} is 50, and the maximum frame rate $f_{max}=30$ fps. To test the accuracy of the P-R-D model, we run the video encoder over the "Foreman" QCIF sequence at 128 kbps and 15 fps for different complexity control parameters (x,y) and measure the corresponding distortion. Figure 4.4 shows the actual distortion function $D(x, y)$. The estimation given by the P-R-D model is shown in Figure 4.5. We can see that model estimation is quite accurate. Simulations over other test videos yield

similar results. For a given bit rate Rand device power supply level, using (4.31) the encoder can find the best configuration of complexity control parameters to maximize the video quality. Figure 4.6, Figure 4.7, and Figure 4.8 show the picture distortion, and the optimal control parameters *{x,y,z}* as functions of the percentage of power consumption for different coding bit rates *R*. Some interesting observations can be made: (1) As the encoder scales down its power consumption, as a percentage of its maximum power consumption level, the video quality degrades. The video encoding automatically changes from high quality motion video coding (when the energy supply is plenty) to still image coding (when the device is running out of energy). (2) At lower bit rates, the ME wins over the PRECODING in power allocation, because the ME is computation-hungry but the PRECODING is bit-rate-hungry; hence, as shown in Figure 4.6, the complexity for the ME is high but the complexity for the PRECODING is low. Figure 4.9 shows the achievable minimum distortion *D* as a function of *R* and the power *P*. Figure 4.10 shows the "Carphone" QCIF video coded at 64 kbps and 15 fps for different power consumption levels. We can see the picture quality degradation is very graceful. We can see that the P-R-D model has direct applications in energy management, resource allocation, and QoS provisioning in wireless video communication, especially over wireless video sensor networks.

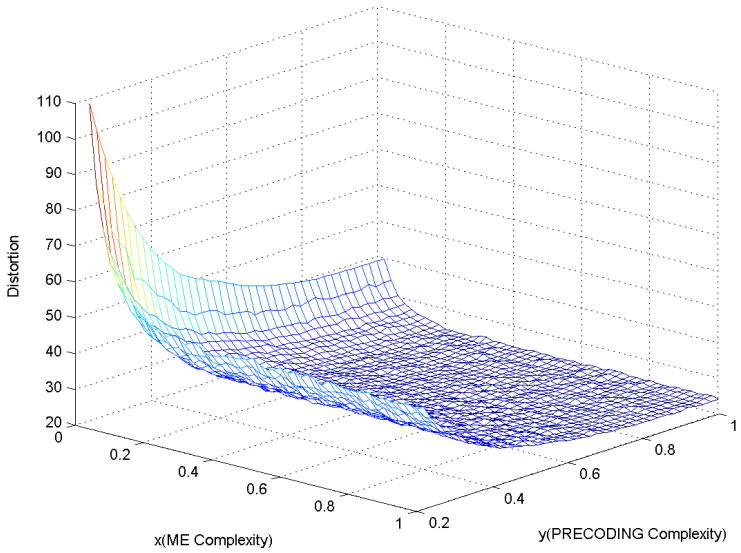


Figure 4.4 Actual complexity-distortion surface D(x,y).

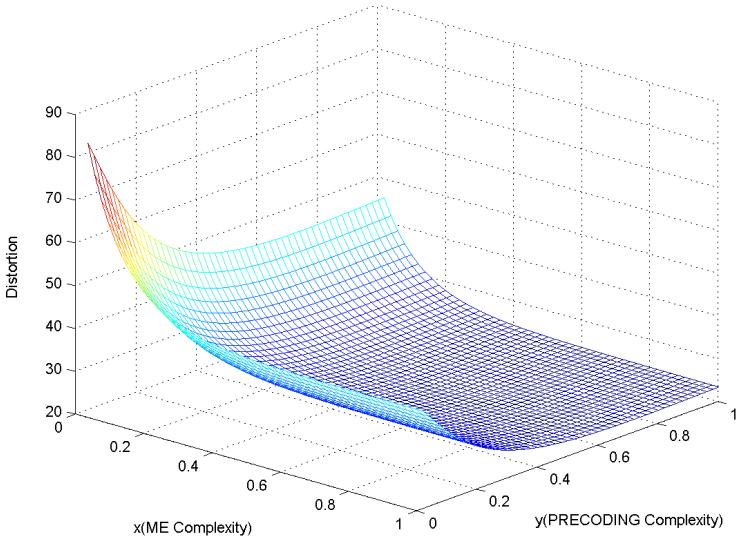


Figure 4.5 The complexity-distortion surface estimated by the P-R-D model.

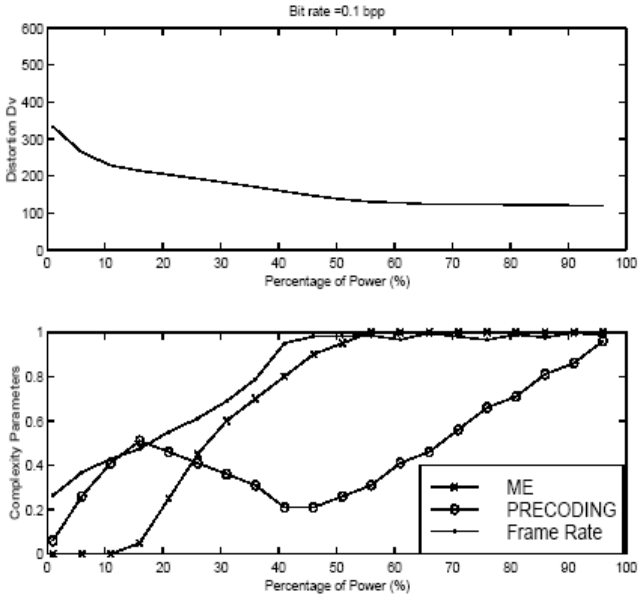


Figure 4.6 R-D optimized power control for the "Football" CIF video at R=0.1bpp.

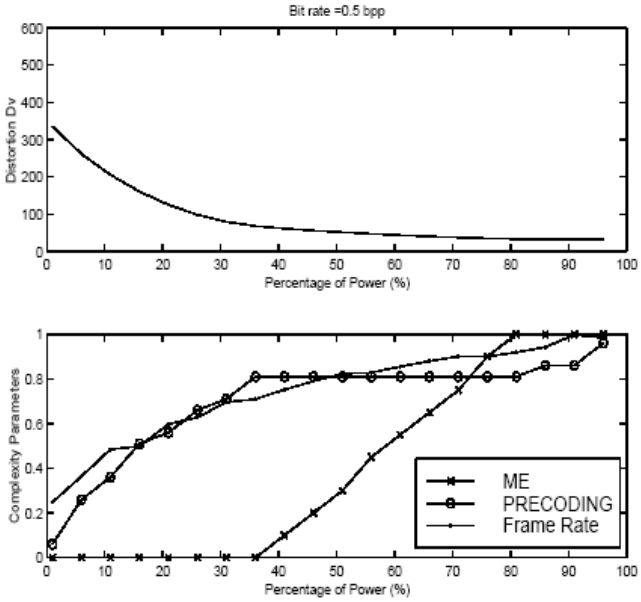


Figure 4.7 R-D optimized power control for the "Football" CIF video at R=0.5bpp.

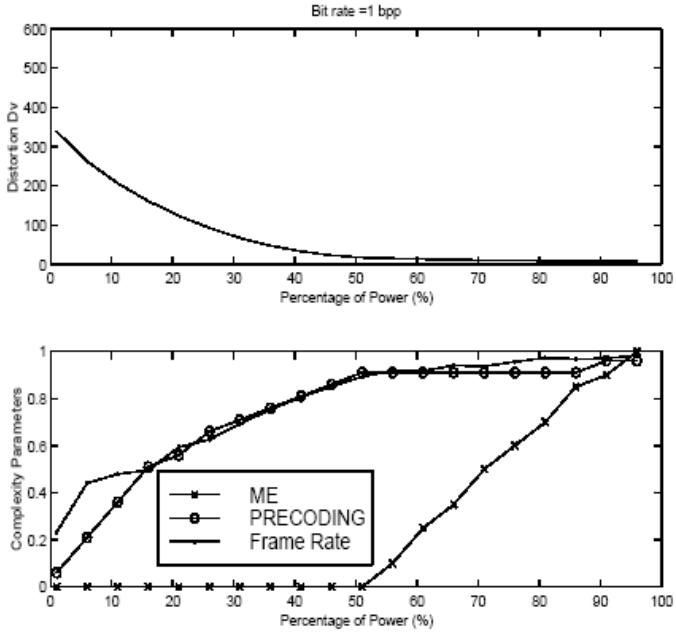


Figure 4.8 R-D optimized power control for the “Football” CIF video at R=1bpp.

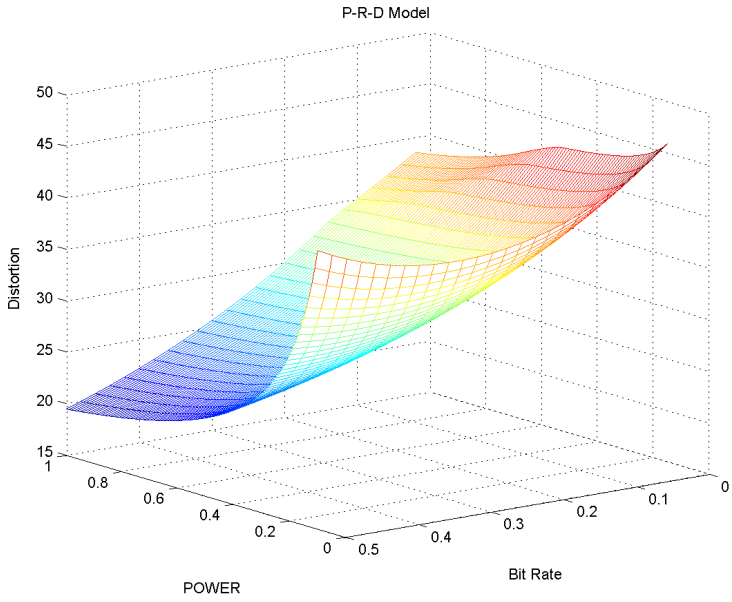


Figure 4.9 The P-R-D Model.

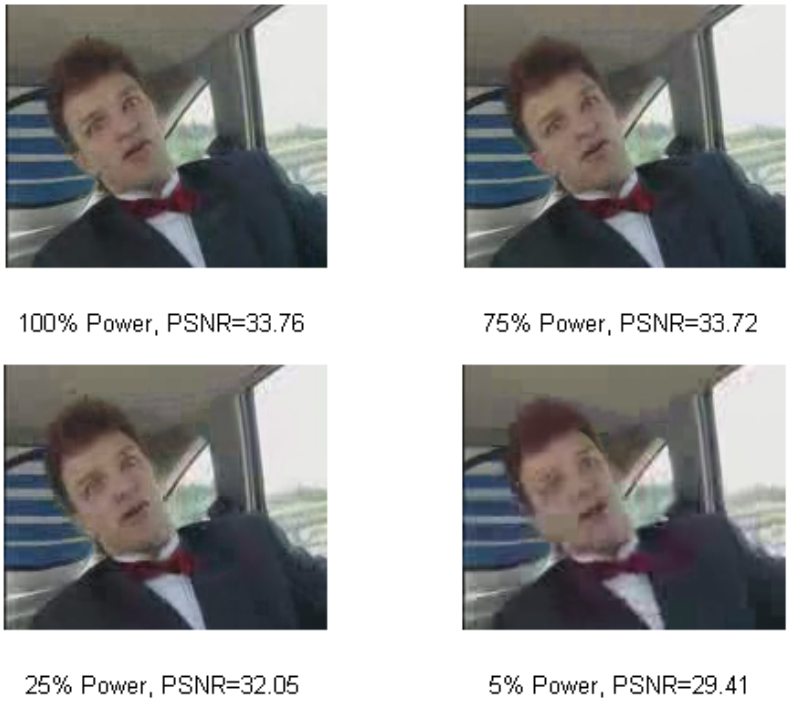


Figure 4.10 The encoded "Carphone" QCIF sequence at 64 kbps and 15 fps for different power supply level.

4.6 Concluding Remarks

In this chapter, we have successfully extended the traditional R-D analysis by considering another dimension, the power consumption, and established the P-R-D analysis framework for video encoding and communication under energy constraints. Using the P-R-D model, given a power supply level and a bit rate, the power-scalable video encoder is able to find the best configuration of complexity control parameters to maximize the video quality. The P-R-D analysis establishes a theoretical basis and provides a practical guideline in system design and performance optimization for wireless video communication under energy constraints.

CHAPTER 5

POWER AND DISTORTION OPTIMIZED VIDEO CODING

5.1 Motivation

Traditional power-conserving video processing techniques usually focus on low-power device and circuit design, such as [8], [11]. Recently, some research works on high-level management of algorithms and architectures have been reported, including power-aware design techniques that attempt to adjust the source-coding parameters to maximize the performance under power dissipation constraints [63], [65] and low-power design techniques that try to lower the intensive complexity of video encoding with or without a desired performance target [5], [67], [87], [88]. In futuristic pervasive video coding, a well designed video encoder should be able to provide high quality of service, which is measured from two perspectives: First, the video quality at a given bit rate must be optimized; Second, it should be able to efficiently utilize its power supply to prolong the battery operating time. In other words, it should be able to achieve low power consumption and video distortion. However, the existing work only focuses on achieving minimal power consumption or minimal video distortion, and the power-distortion optimization problem has received little attention.

In power-distortion optimized video coding, the coding complexity needs to be taken into account since it affects both the power consumption and the video distortion.

This in turn requires developing an appropriate R-C-D model to study and understand the interaction and tradeoffs between rate, complexity, and distortion. An appropriate model helps to provide valuable insights into the power-distortion optimization problem. Some tradeoffs of power and complexity have been addressed. In [82], using the percentage of INTRA coded macroblocks as the complexity parameter, a distortion-rate (DR) model with six model parameters is presented to describe the video coding DR performance for error-prone video transmission. This DR model is used in [67] for adaptive minimization of the total power consumption of wireless video communications subject to a given end-to-end distortion. In [40], a distortion–computation function is defined in a systematic way to study the complexity-distortion relationship. Research works on R-C-D modeling are also reported in [63], [79]. In our previous work in [43], based on the analysis of power consumption, bitrate, and distortion behavior, a video system that is able to minimize the distortion under a given power consumption constraint is developed. However, the little reported works are either ad hoc, in that they depends on the coding behavior of a specific coding algorithm, or they propose complex models so that estimation of model parameters might become a formidable task. Up until now no general R-C-D model has been proposed.

Unlike the work in CHAPTER 5 in which the P-R-D model is basically a distortion minimization problem constrained by power consumption. This chapter addresses the issues of power-distortion optimized video coding. Our focus is on techniques for efficiently utilizing the energy supply while preserving desirable video

quality. In fact, we can see later that the problem in CHAPTER 5 is a sub-problem of the problem addressed in this chapter. To accomplish this goal, we first formulate a multiple objective optimization problem to model the behavior of power-distortion optimized video coding. Then, as the first attempt, we develop a R-C-D model to describe the general R-C-D behavior of video coding. Based on the proposed model and the analysis of power consumption, optimization strategies are developed to achieve good video quality while maximizing the battery service life.

5.2 The Power-Distortion Optimization Problem

5.2.1 Parametric Video Encoder Design

The coding procedure can be simply decomposed into three consecutive parts: (1) ME/MC; (2) Mode selection; (3) Entropy coding. The first two parts produce the quantized transform coefficients. The third part is responsible for converting the symbols of quantized transform coefficients into a standard compliant bitstream. Parameterization can be introduced into these parts, since video coding standards only define the bitstream syntax, or decoder operation, parametric video coding causes no conflict. Parametric ME can be achieved by controlling the motion estimation precision [21], the size of the motion vector (MV) search window [85], or the number of search points during the searching for the MVs [43]. Parametric mode selection can be achieved by controlling the λ_{PRE} parameter, which is the fraction of non-SKIP MBs in the video frame [43], or limiting the number of available coding modes, or the INTRA ratio parameter, which is the fraction of INTRA MBs in the video frame [82], [67]. By

pre-setting/ skipping the coding modes, a great amount of coding operations involved in mode decision can be saved. Parametric control can also be applied on DCT and quantization, such as the technique proposed in [70]. Since the output bitstream must be conformed to bitstream syntax defined by the specific video standard, parameterized coding is not performed in the component of entropy coding.

Let $C = \{c_1, c_2, \ldots, c_N\}$ be the parameter set of the parametric encoder where N is the total number of parameters. For example, we may have parameter c_1 for ME, which is the size of the search window, and c_2 for mode selection, which is the INTRA ratio parameter. Without loss of generality, each element c_i is normalized to lie between 0 and 1, measuring the level of coding complexity, wherein 0 and 1 corresponds to no and full coding complexity, respectively. The larger the value, the higher the coding complexity is. In the following, c_i is referred as the complexity parameter.

5.2.2 The Problem

The complexity parameters affect the compression performance in terms of power consumption and video distortion. Let $P(R,C)$, $D(R,C)$ denote the power consumption and video distortion with parameter set C at coding bit rate R; detailed analysis will be given in the next section. In power-distortion optimized video coding, the objective is to find a parameter set that minimizes both the power consumption and the video distortion, which can be formulated as a multiple objective optimization (MOO) problem, given by

$$\min\begin{bmatrix} P(R,C) \\ D(R,C) \end{bmatrix}, \quad s.t. \quad P(R,C) \le P_c, \tag{5.1}$$

where P_c is the power supply constraint for video coding. We assume P_c is attainable either through low-level circuits design or pre-determined by system-level power allocation. This assumption can be worked around by using techniques proposed in the literature, such as estimating the remaining power using microcontroller [11] and the "state-of-charge" techniques [14]. For simplicity, we only consider the case that bit rate is given as a constant and controlled by a rate control scheme.

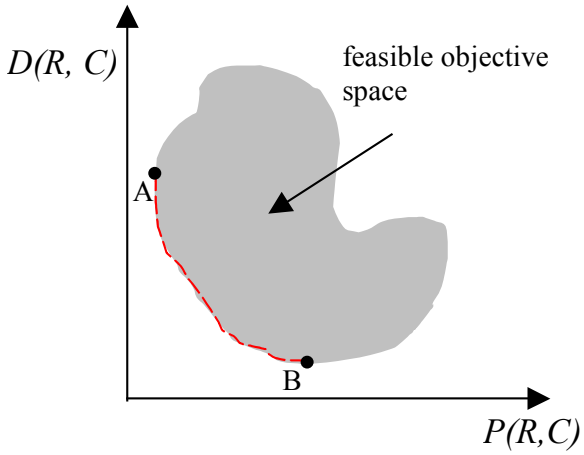


Figure 5.1 Illustration of the MOO problem.

In general, higher coding complexity results in smaller distortion, but consumes more energy. In contrast, lower coding complexity has lower power consumption, but at the expense of larger distortion. Thus, the objective functions in (5.1) are incommensurate and in conflict with one another with respect to their minimum goals. For such a MOO problem, there is no unique solution [28]. Figure 5.1 illustrates the concept of the MOO problem with conflict objective functions. Curve *AB* is the pareto-optimal frontier, along which no further improvement can be done on power consumption *P* or video distortion *D* without sacrificing the other one. Hence, the goal

is to search for a tradeoff between the distortion and the power consumption to ensure a satisfactory design.

5.3 Problem Analysis and Optimization Strategy

5.3.1 Power Consumption Analysis

To model the power consumption of a parametric video encoder, ideally we could perform a theoretical analysis at the instruction level. However, this approach depends upon the platform on which the video encoder is implemented and how the encoder is implemented. Many factors, such as the instruction set, supply voltage and CPU frequency, would influence the analysis. To present the main concept, we model the power consumption at higher levels of abstraction (algorithmic and architectural levels).

The video coding operations are decomposed into two parts: 1) Residual error prediction and transform; 2) Entropy coding. The first part includes coding operations such as ME, DCT, IDCT, MC, etc, wherein the complexity parameter is usually introduced, as discussed in the previous section. Obviously, the energy consumed by this part depends on the embedded complexity parameters. The energy consumption of entropy coding is approximately proportional to the coding bit rate R [69]. We model the energy consumption (with units of Joules) of encoding one frame by

$$E_f = E_0 + E_r \cdot R + Ec, \tag{5.2}$$

where E_0 accounts for miscellaneous energy overheads, such as energy cost for I/O, and E_r is a linear factor constant. Ec is the energy consumption of the parametric coding

operations. In this work, we consider *Ec* as the summation of the energy consumption of individual modules:

$$E_c(C) = \sum_{i=1}^{N} E_i c_i \,, \tag{5.3}$$

where E_i is the energy consumption for module *i*. Let f_s denote the coding frame rate (frames/second), the power consumption (with units of Watts) is given by

$$P(R, C) = f_s \cdot E_f = f_s \cdot (E_0 + Ec(C) + E_r \cdot R) = P_0 + Pc(C) + P_r R. \tag{5.4}$$

Note that changing the complexity parameters and bit rate affects the miscellaneous energy overheads. We observe, however, even if efficient models of energy overheads are available, the dependence is not obvious making a comprehensive compositional model potentially complex. Thus, in this work, we ignore this effect and E_0 is treated as a constant. Because the complexity parameters are normalized values and bounded by 1, *P* is also bounded. When the video encoder runs at full complexity ($c_i = 1$), *P* is the maximum, denoted by P_{max}.

5.3.2 R-C-D Modeling

For a parametric video encoder, *D* is not only a function of *R*, but also the complexity parameters. To model the R-C-D behavior, we begin the analysis by considering only one complexity parameter *c* in the video encoder. In this case, *D(R, C) = D(R, c)*. Two implementations of parametric video coding are investigated. In the first implementation, we use the search range of full search ME as the complexity parameter with maximum value 15. When the search range decreases, the complexity decreases since we have less search points for MV search. In the second implementation, we use

the INTRA refresh rate as the complexity parameter with maximum value 10. In this approach, if the consecutive times of one MB being coded as an INTER MB are bigger than the INTRA refresh rate, this MB will be forced as INTRA coded. As the refresh rate decreases, the complexity of the encoder decreases since the MBs are more frequently coded as INTRA MBs for which ME is not required. In Figure 5.2 and Figure 5.3, the R-D curves of frame 0~99 of "Stefan" sequence at different constant values of c ($c_1 < c_2 < c_3 < c_4$) are plotted. Note that c is a normalized value.

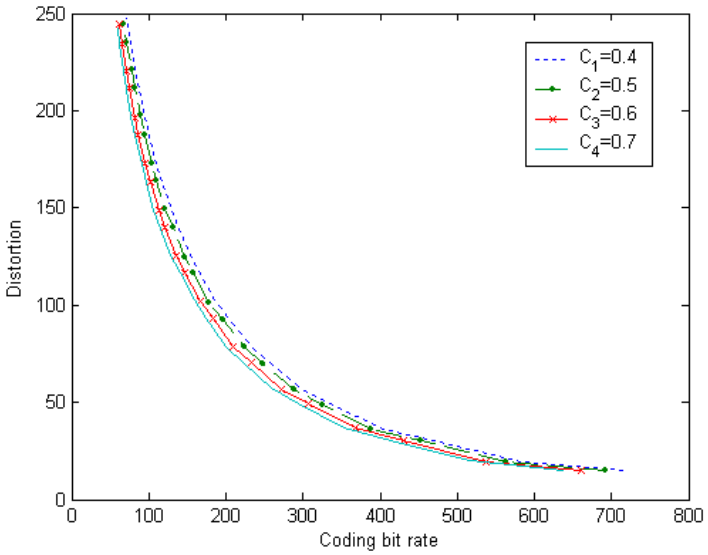


Figure 5.2 R-D curves of "Stefan" sequence with ME search range control.

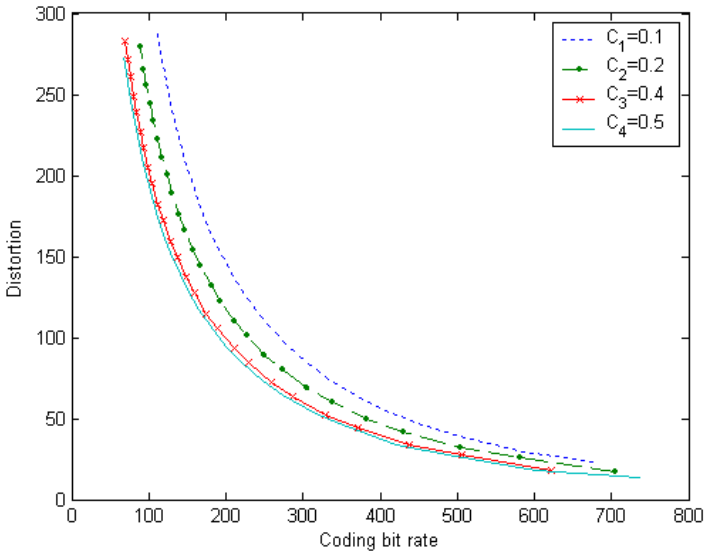


Figure 5.3 R-D curves of "Stefan" sequence with INTRA refresh rate control.

As can be seen from Figure 5.2 and Figure 5.3, the R-D performance deteriorates as c decreases. Based on the assumption of independent identically distributed (*i.i.d.*) memoryless source for the quantized transform coefficients, typical R-D model under the mean square distortion criterion is

$$D(R) = \varepsilon^2 \cdot \sigma_x^2 \cdot e^{-\alpha R}, \tag{5.5}$$

where ε^2 is a source dependent parameter equal to 1 for uniform distribution, 1.4 for Gaussian distribution, and 1.2 for Laplacian distribution. σ_x^2 denotes signal variance. The parameter α equals to 1.386 for uniform, Gaussian and Laplacian distribution [41]. Similar to the R-D model, but taking into account of the impact of c on the R-D performance, we model the R-C-D behavior with

$$D(R, c) = \varepsilon^2 \cdot \sigma_x^2 \cdot c^{-\beta} \cdot e^{-\alpha R} = \gamma \cdot c^{-\beta} \cdot e^{-\alpha R},\ 0<c\leq 1,\ \alpha > 0 \text{ and } \beta > 0, \tag{5.6}$$

where α, β, and γ are model parameters and can be obtained by using linear regression techniques. When we have constant complexity, the R-C-D model becomes the classical R-D model. It is worth to mention that different implementation of parametric video coding has different impact on the video distortion. For example, the video distortion is less sensitive to parametric ME control than to parametric DCT/Quantization control where the distortion mainly comes from. Parameter β accounts for this difference. For a specific video encoder, it is possible to have a more accurate (but potentially more complex) R-C-D model through theoretic R-D analysis. However, this theoretic R-C-D model is only valid for the specific video encoder, which limits its application. In contrast, the proposed R-C-D is a general model. As we show in the following, it remains valid for different video encoders. The simplicity of the model significantly increases its usability and provides valuable insights into the R-C-D tradeoffs in power-distortion optimization.

We empirically evaluate the accuracy of the proposed R-C-D model, using the ME search range control and INTRA refresh rate control. For each implementation, we vary the complexity parameter and the quantization factor to obtain the output bit rate and distortion. We perform data fitting by minimizing the sum of squared MSE differences between the model and the measured data. Figure 5.4 ~ Figure 5.7 present the results of the actual R-C-D surfaces and the results estimated from the model. One may notice that the actual R-D relation are more heavily damped at low bit rates than the proposed model, this is because the assumption of *i.i.d.* memoryless source is not typically found in real sequences. We can see that different implementations of

parametric video coding have significantly different forms of R-C-D surfaces. For example, the surface of complexity control using ME search range is more flat than the one using the INTRA refresh rate. Nevertheless, the proposed R-C-D model still fits the actual data well. Simulations over other test video sequences yield similar results. The average prediction accuracy is given in Table 5.1. The proposed model is accurate enough to appropriately describe the R-C-D behavior of video coding.

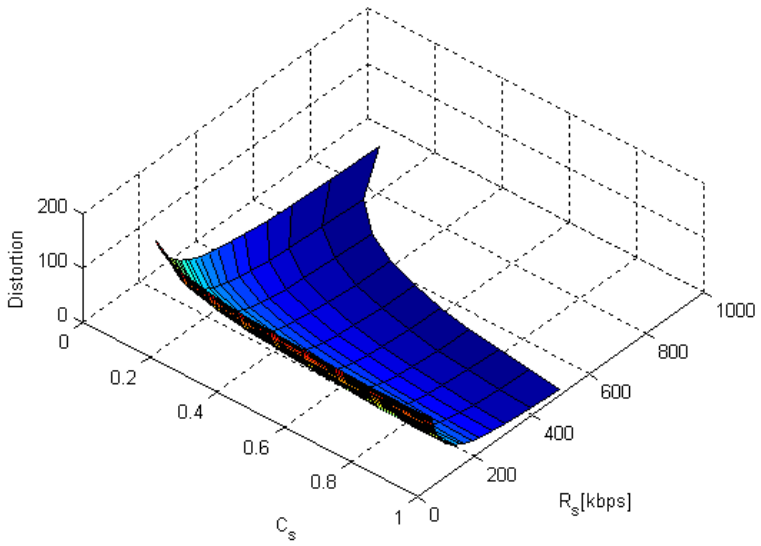


Figure 5.4 R-C-D surface using INTRA refresh-rate control for “Foreman” QCIF.

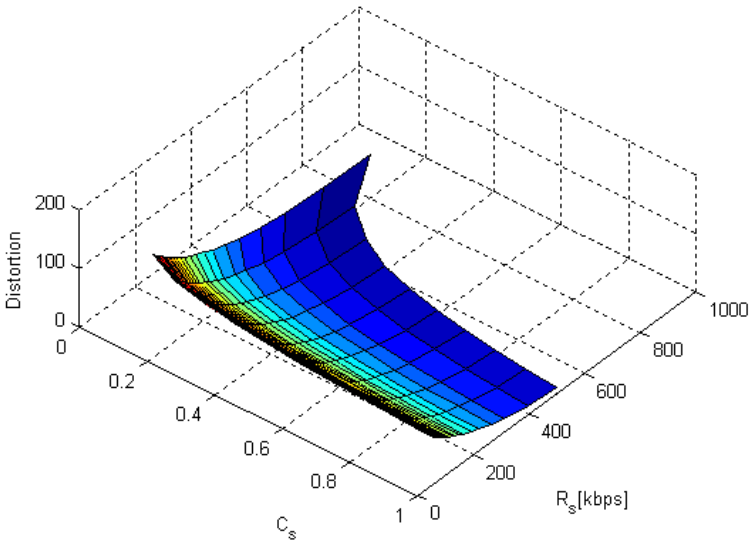


Figure 5.5 Data generated by the proposed R-C-D model using INTRA refresh rate control for "Foreman" QCIF sequence.

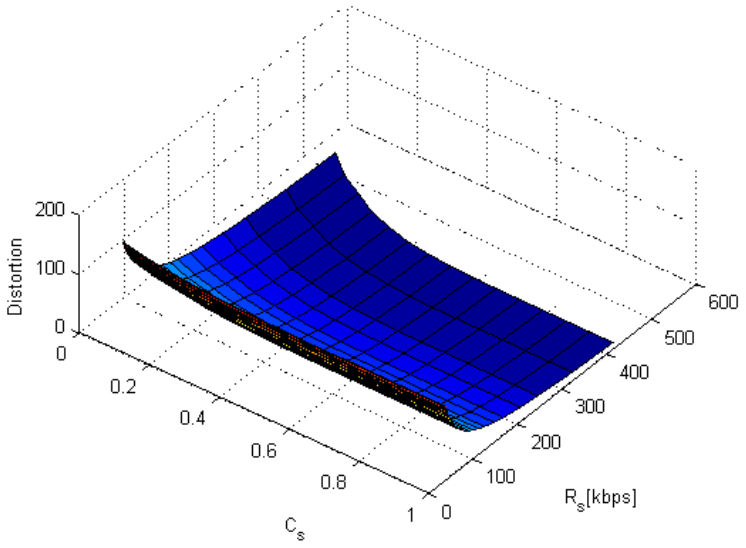


Figure 5.6 R-C-D surface using ME search range control for "Foreman" QCIF.

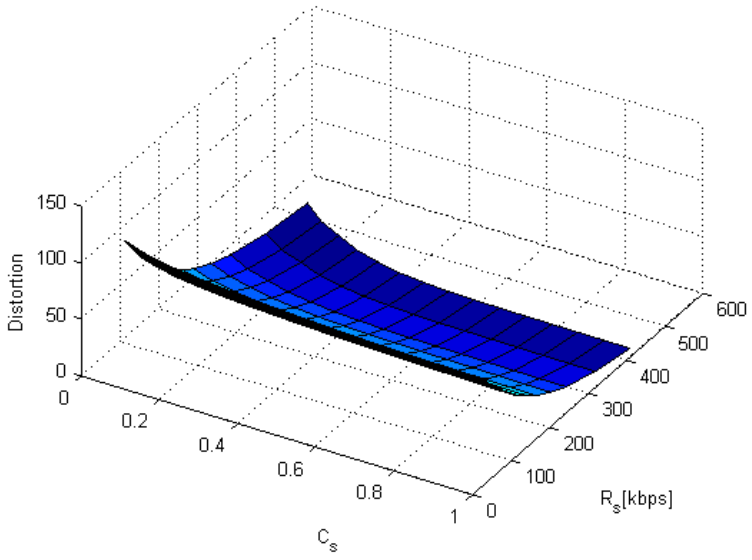


Figure 5.7 Data generated by with the proposed R-C-D model using ME search range control for "Foreman" QCIF sequence.

Table 5.1 Average Prediction Accuracy of the R-C-D Model

Parametric Video Coding	Foreman	Carphone	Akiyo
ME search range control	83%	82%	89%
INTRA refresh rate control	82%	81%	80%

Now we consider the case when the parametric encoder has multiple complexity parameters: $C=\{c_1, c_2, \ldots c_N\}$. Since the encoder can be regarded as a nonlinear system, the overall distortion is not simply a linear addition of the individual effects of c_i. As discussed above, the complexity parameters are usually separately applied to different coding modules. We observe that the effects of various complexity parameters can be described in a separable manner. Based on the observation, we model the R-C-D using

$$D(R,C) = \gamma \cdot e^{-\alpha R} \prod_{i=1}^{N} c_i^{-\beta_i}, 0 < c_i \le 1, \alpha > 0, \beta_i > 0, \tag{5.7}$$

where N is the number of complexity parameters. A larger model parameter β_i indicates c_i has more contribution/effect on the overall distortion.

5.3.3 Optimization Strategy

Because the energy drawn from a battery is not always equivalent to the energy consumed in device circuits, understanding the battery discharge behavior is essential for optimal system design. Figure 5.8 gives an example of the discharge characteristic of the Lithium-ion battery [76], which is used widely in today's mobile devices because of its high energy density and capacity.

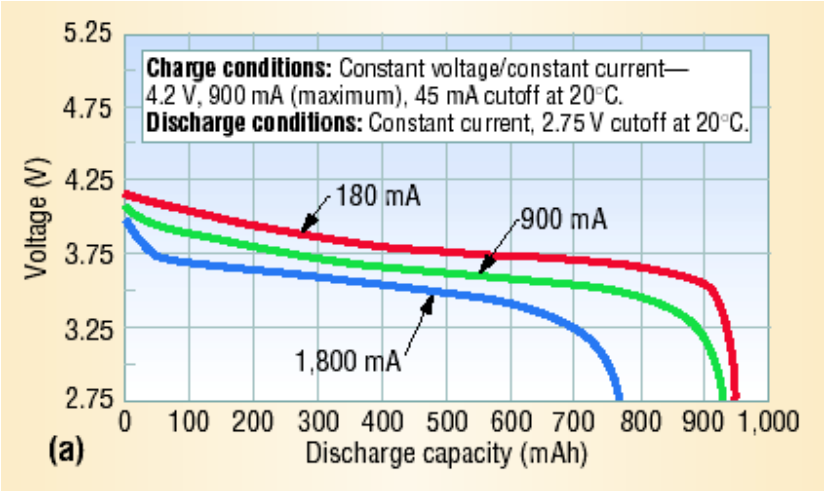


Figure 5.8 Lithium-ion battery discharge characteristics.

Observation 1: As the battery discharges, its voltage drops. There is an inflexion point on the discharge curve, after this point the power will run out quickly. In this case, the video quality will degrade quickly due to insufficient power supply.

Observation 2: The effective battery capacity is increased if the average discharge current from the battery decreases, which suggests that reducing the discharge current, i.e., lowering the complexity is essential for battery lifetime extension.

According to these observations, we have different preferences for the distortion versus power consumption during the whole battery lifetime. Thus, we apply the constraint-oriented strategy to solve the above MOO problem. In this strategy, one objective function is used as the main objective and the other is treated as the secondary objective.

5.3.3.1 Distortion Preference

When we do not have enough power supply to perform full complexity encoding, i.e., $P_c < P_{max}$, we substitute problem (5.1) by

$$min\ D(R, C),\ s.t.\ P(R, C) < P_c, \tag{5.8}$$

which is a distortion optimization problem with a power consumption constraint. Using the Lagrange multiplier method to solve the problem, we have the Kuhn-Tucker conditions as

$$\frac{\partial D}{\partial c_i} + \lambda \frac{\partial P}{\partial c_i} = 0, \tag{5.9.a}$$

$$\lambda \geq 0, P(R,C) \leq P_c, \text{and } \lambda(P(R,C) - P_c) = 0, \tag{5.9.b}$$

where λ is the Lagrange multiplier controlling the P-D tradeoffs. From the last condition (5.9.b), either $\lambda = 0$ or $P(R, C) = P_c$. If $\lambda = 0$, from (5.9.a), we should have

$$\frac{\partial D}{\partial c_i} = 0, \tag{5.10}$$

which is conflict with the following equation derived from the R-C-D model:

$$\frac{\partial D}{\partial c_i} = -\gamma\beta_i c_i^{-\beta_i - 1} \prod_{j=1, j\neq i}^{N} c_j^{-\beta_j} e^{-\alpha R} < 0 \text{ , } i \neq j. \tag{5.11}$$

As a result, we have

$$P(R, C) = P_c. \tag{5.12}$$

Remark 1: In the distortion-preferred optimization problem, the optimal solution is at the boundary of the constraint where the power consumption is maximized to the given upper bound. That is, the video encoder has to consume all the available energy to get the minimal distortion.

The solution can be obtained by solving (5.9.a) and (5.12). As an example, suppose we have two complexity parameters, c_1 and c_2, we have

$$D = \gamma c_1^{-\beta_1} c_2^{-\beta_2} e^{-\alpha R}, P = P_0 + P_1 c_1 + P_2 c_2 + P_r R. \tag{5.13}$$

The solution is

$$c_1 = \frac{P_c - P_0 - P_r R}{P_1 (1 + \frac{\beta_2}{\beta_1})}, c_2 = \frac{P_1 \beta_2 c_1}{P_2 \beta_1}. \tag{5.14}$$

When only one complexity is used, the solution can be easily obtained without solving the unconstrained problem. The optimal complexity parameter is given by

$$c_{opt} = (P_c - P_0 - P_r R) / P_1. \tag{5.15}$$

5.3.3.2 Power Consumption Preference

When $P_c \geq P_{max}$, to preserve power, we substitute problem (5.1) by

$$min\ P(R, C), s.t.\ D(R, C) \leq D^*, \tag{5.16}$$

where D^* is a given upper bound distortion. Notice that different from (5.1), here the power constraint is released. Analysis on the Kuhn-Tucker conditions results in a similar conclusion:

Remark 2: minimal power consumption is achieved when the video distortion is maximized to the upper bound value, in other words, when

$$D(R, C) = D^*. \tag{5.17}$$

Figure 5.9 gives one example of how the upper bound distortion affects the solution. The power consumption is measured by the percentage of power compared with that is consumed by running at full coding complexity.

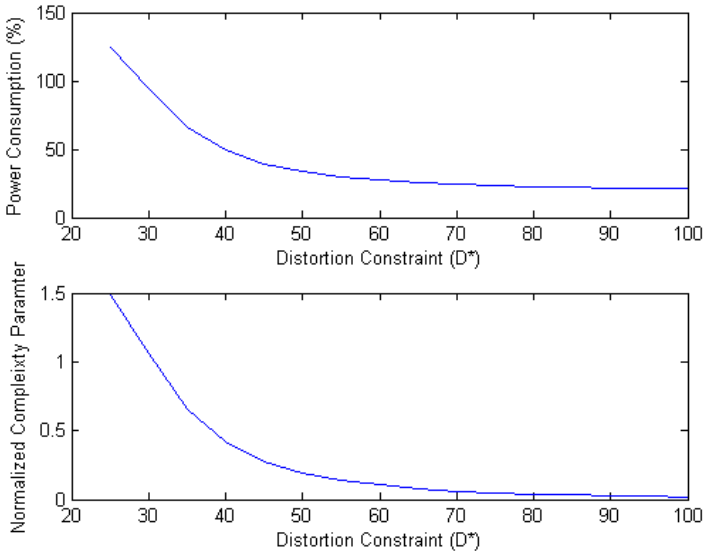


Figure 5.9 P-D curves for "Foreman" QCIF sequence using INTRA refresh control under different distortion constraints.

From Figure 5.9, we notice that as the distortion constraint increases, the video encoder can run at lower coding complexity for smaller power consumption. Also notice that as the expected distortion decreases constraint more, bigger complexity

parameters, and thus higher power consumption is required. In some cases, it is even not feasible, as shown in the figures, the complexity parameter bigger than 1 and Power Consumption higher than 100%.

Considering the human being can tolerate some video distortion, this degradation in perceived video quality may be negligible with respect to the improvement in power saving. As the curve shows, this improvement in power consumption by increasing the distortion constraint will get saturate. Therefore the selection of an appropriate distortion constraint is crucial for power consumption saving. However, the constraint of distortion is very subjective and application dependent, which makes it quite difficult to pre-determine the distortion constraint before encoding the sequence. It is desirable to have a dynamic control to attack this challenge.

Remark 3: From (5.11), each complexity parameter has different influence on the overall distortion according to their importance, reflected by their model parameter β_i. For a same amount of complexity change, the larger the value of β_i, the more changes in the distortion.

Remark 4: For a given amount of complexity change, the smaller the value of c_i, the larger the decrement of distortion is. Thus, as the encoder slows down, (c_i becomes smaller), the distortion becomes more sensitive to the change of complexity.

Based on *Remark 3* and *Remark 4*, instead of pre-determining the distortion constraint, we apply a progressive control approach. Each parameter is adjusted individually according to their importance. Note that for a given bit rate, the minimum

distortion can always be achieved by running at full complexity from *Remark 1*. We start running the video encoder at full coding complexity, and progressively lower the complexity to preserve power. According to *Remark 4*, it is desirable to promptly increase the complexity when the video quality begins to deteriorate. The advantage of the progressive control is that we do not need accurate model parameters to solve the minimization problem. Also, since a predefined upper bound distortion is not required, this approach is adaptive to various coding content.

5.4 Power and Distortion Optimized Video Coding

In this section, based on the optimization strategies presented in the previous section, we demonstrate how to achieve power-distortion optimized video coding in a practical parametric video encoder, developed in CHAPTER 5. We use ME and model selection as the candidates for complexity control, since they significantly affect both the video distortion and power consumption. For the ME module, the computational complexity is determined by the number of SAD (sum of absolute difference) computations associated with each frame, denoted by c_1. The available number of SAD computations is dynamically allocated throughout the frame among the MB's according to the motion intensity. For the mode selection control, the fraction of MBs being coded as none zero MB (NZMBs) is used as another complexity parameter, denoted by c_2. Remember that when a MB is code as a zero MB, coding operations including DCT/IDCT quantization/dequantization are skipped. For more detail, refer to CHAPTER 5. Different adjustments are applied on c_1 and c_2.

5.4.1 Progressive Complexity Adjustment

5.4.1.1 Scene Change

After applying the ME and mode selection control, which will be described below, the encoder might be in "sleep" (if the scene has been idle for a while), therefore it is necessary to "wake up" the encoder when with rapid motion or sudden scene change. A variety of techniques have been proposed to perform detection of scene changes. Considering the requirement of low computational complexity in the research, we use the percentage of INTRA coding MBs to detect scene change. This approach has low computational overhead, but still have pretty good performance. When scene change occurs, the video coding will restart at the highest complexity level in order to re-understand the content and choose another optimal parameter set.

5.4.1.2 Greedy Motion Estimation

We propose a greedy adjustment on the complexity scalable ME. We start with the highest complexity parameter value and keep reducing the value until we obtain a performance level that is no longer acceptable. Because of its simplicity, the sum of SADs of all the MBs is used as the distortion measure.

Note that the process of ME can be simply considered as a sequence of SAD computations. For each MV candidate, one SAD value is calculated and compared with the previous minimal SAD. A smaller value of SAD indicates a better MV. As the search for the optimal MV continues, more SAD computations are required and smaller SAD may be obtained. Denote SAD_m and $SAD_m^{'}$ the last SAD value and the second last SAD during the MV search for the m^{th} MB. At frame i, let

$SSAD_i = \sum_m SAD_m, SSAD_i^{'} = \sum_m SAD_m^{'}$ be the sum of SAD and the sum of second last SAD respectively. Clearly, $SSAD_i$ needs more computation than $SSAD_i^{'}$. If $SSAD_i$ is smaller than $SSAD_i^{'}$, it implies we may get smaller distortion if we increase c_1 at frame *i+1*. Let c_e^i, c_a^i, denote the estimated and actual complexity parameter for frame *i*, respectively. The greedy ME control is given in Figure 5.10. We exponentially increase the complexity when we can get a performance improvement.

```
if (SSAD_i –SSAD_i' < Thresold)
/* We may get improvement by increasing the complexity */
        N++;
        if (c_a^i > c_e^i)
            c_e^{i+1} = c_e^i + 2^N · (c_a^i − c_e^i)
        else
            c_e^{i+1} = c_e^i
else
        N = 0;
        c_e^{i+1} = min(c_a^i, c_e^i)
```

Figure 5.10 Greedy ME control.

The *threshold* in Figure 5.10 controls the C-D tradeoff. In the experiments, we use the extreme value *threshold = 0*. Figure 5.11 shows the results of greedy ME complexity control for three different sequences, representing motion characteristic from low motion to high motion respectively. The "Akiyo" sequence is a typical head-and-shoulder sequence and has very little motion. The camera is assumed to be stationary. From Figure 5.11(a) we can observe that the ME complexity parameter c_1 is gradually lowered to preserve power and finally stays in constant when the encoder only consumes computation on the active region (the news reporter). The "Foreman"

sequence has medium motion with camera motion. In Figure 5.11(b), the encoder resets the complexity parameter to the highest value when a scene change is detected and starts another greedy search. “Stefan” represents a typical sports sequence with fast motion. In “Stefan”, from the 180th frame, the camera moves very quickly to focus on the sportsman. The proposed control is able to increase the complexity parameter promptly when the scene switches to such a fast motion, as shown in Figure 5.11(c).

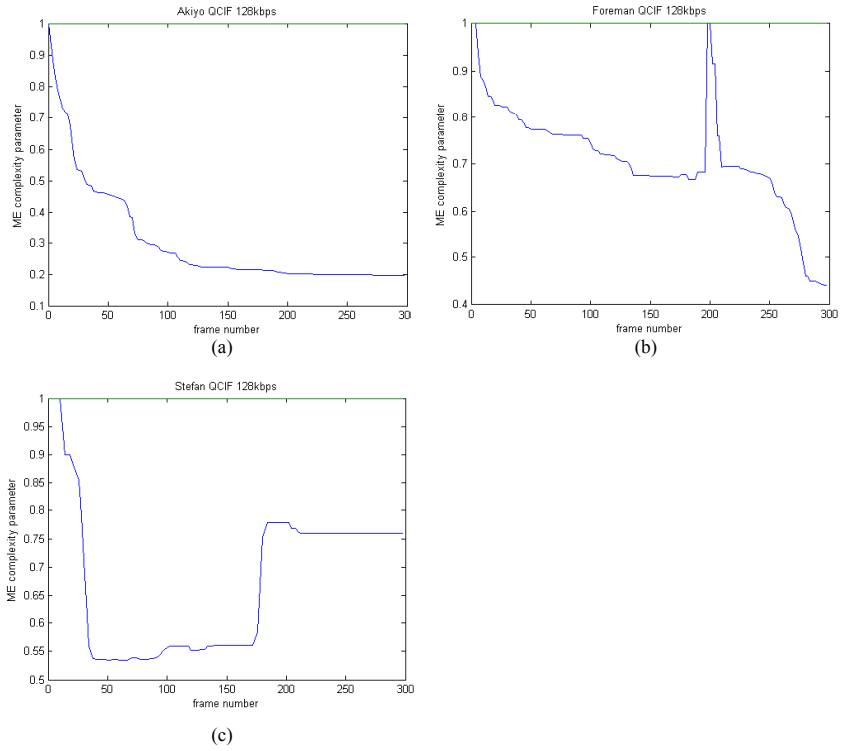


Figure 5.11 Results of greedy ME complexity control. (a) Akiyo QCIF; (b) Forman QCIF; (c) Stefan QCIF.

5.4.1.3 Optimal Mode Selection

Offline simulation shows that the model parameter of model selection is much bigger than that of ME, i.e., $\beta_2 > \beta_1$. This is valid because in video coding, the

quantization process, affected by the procedure of mode selection, mainly causes the distortion. Instead of using progressive control as ME, we try to find the optimal value for c_2.

In a block based video encoder, the residual error after motion compensation is transformed into the frequency domain using certain type of transform, such as DCT. After that, the transform coefficients are quantized by a predetermined quantization parameter for further entropy compression. We denote *r, R* as the residual error and the transform coefficients respectively. *r* can be the input MB for INTRA coding. The transform can be described by

$$R = ArA^T, \tag{5.18}$$

where *A* is the transform matrix. The distribution of the residual error can be approximated by a Laplacian distribution with zero mean and a separable covariance [72]:

$$r(m,n) = \sigma_r{}^2 \rho^{|m|} \rho^{|n|}, \tag{5.19}$$

where *m* and *n* are the horizontal and vertical distances between two pixels respectively, σ_r is the variance of the residual errors. ρ *(*$|\rho|<1$*)* is the correlation coefficient. Typical value of ρ ranges from 0.6 to 0.75. Let *L* denote the matrix of the correlation coefficients, given by

$$L = \begin{bmatrix} 1 & \rho & \rho^2 & & \rho^M \\ \rho & 1 & \rho & \cdots & \\ \rho^2 & \rho & 1 & & \\ & \vdots & & \ddots & \\ \rho^N & & & & 1 \end{bmatrix}. \tag{5.20}$$

Let Q be the quantization parameter and EM be an estimation matrix, of which the element $EM(u,v)$ is given by

$$EM(u,v) = \frac{K \times Q}{n \times \sqrt{[ALA^T]_{u,u}[ALA^T]_{v,v}}}, \tag{5.21}$$

where K is a constant and n is a confidence parameter. We can estimate the probability of $R(u, v)$ being zero by:

$$SAD < EM(u,v), \tag{5.22}$$

where SAD is the SAD value after ME. When (5.22) is satisfied, $R(u,v)$ will be quantized to zero with probability 68%, 94%, and 99% with n = 1, n = 2, and n = 3 respectively [72].

Since the DC coefficient dominates the transform coefficients, when the quantized DC coefficient becomes zero, we can assume that all the quantized transform coefficients are zero. This assumption will not influence the reconstructed video quality apparently, because the human eyes are more sensitive to low frequency coefficients. Therefore, if $SAD < EM(0,0)$, then $R(0,0)$, which is the DC coefficient, all the quantized transform coefficients are zero, which implies the MB will be a zero MB. After doing this for each MB, we can calculate the percentage of zero MBs and determine c_2. Being able to determine c_2, the number of complexity parameter is reduced to one, which can be calculated by (5.15). This significantly reduces the complexity of solving the distortion-preferred optimization problem, as discussed in the previous section.

5.4.2 System Architecture

Figure 5.12 shows the architecture of the proposed video coding system, which includes four major modules: the "parametric coding modules", the "analysis and adjustment" module, the "energy monitoring" module, and the "bit allocation" module. In the system, the coding modules demanding high computational power are complexity scalable and can be adjusted by the embedded parameter c_1 and c_2. The "analysis and adjustment" modules model the R-C-D behavior of the system and apply the progressive control, aiming to determine a control parameter set for the coding modules. The "energy-monitoring" module provides the current power supply of the system, i.e., the power constraint of this power sensitive platform. The "bit allocation" module determines the number of bits to encode the next frame or group of frames according to the available bit rate determined by the transmission bandwidth.

The optimized video encoder operates as follows:

Step 1. Estimate the model constants of (5.4) from theoretical analysis or experimental profiling.

Step 2. Complexity parameters adjustment: apply the greedy ME control to the ME module. The SAD value is used as the video quality measure. We calculate the complexity parameter c_2 for mode selection. The model parameters α, γ, β_1, and β_2 are estimated from the statistics of previous frame using linear regression. If scene changes occur, we restart the encoding at full complexity.

Step 3. Check the constraint of power consumption. When distortion-optimized is preferred, get the complexity parameters using (5.15).

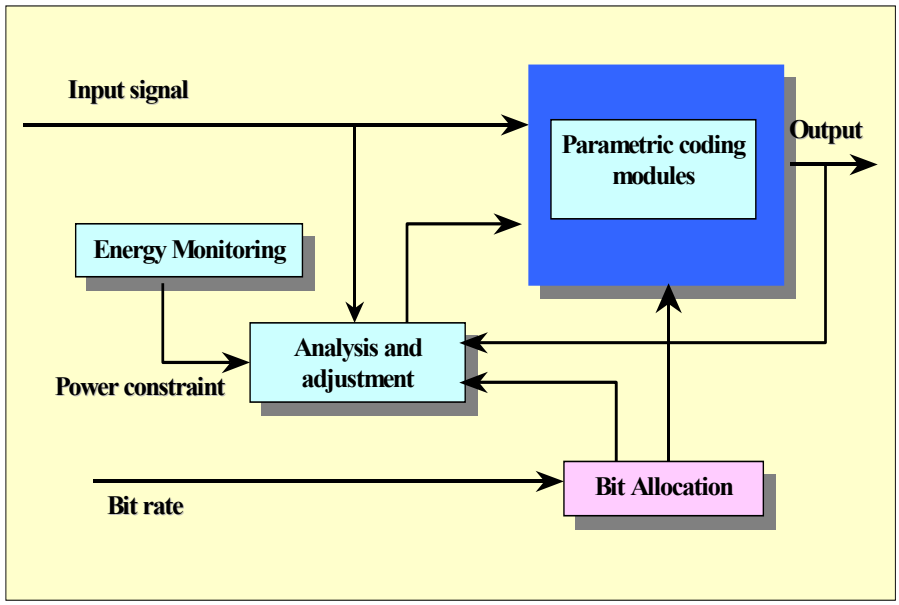


Figure 5.12 Power-distortion optimized video encoder.

5.5 Experimental Results

In this section, we evaluate the performance of the power-distortion optimized vide coding. The public domain H.263+ encoder is tested in the experiments. The approach is generally applicable to other standards and similar performance is expected. The video distortion is measured by peak signal to noise ration (PSNR). To compare with the traditional fast video coding approach, the fast ME method in the reference software is tested in the experiments, referred as the FastME approach in the following. The TMN8 rate control algorithm is used. In the simulations, the power consumption is measured using the linear model, given by (5.2). The complexity ratios are obtained through run-time complexity profile analysis. The maximum ME complexity

corresponds to 50 search points for each MB and therefore the full search ME will use a ME complexity parameter much higher than 1.

Figure 5.13, Figure 5.14, and Figure 5.15 illustrate the comparisons of reconstructed video quality and power consumption for the tested sequences. We can observe that the proposed PDO (power-distortion optimization) video coding achieves significant power saving while maintaining similar video quality compared with the referenced video encoder. The proposed method has much better performance than the referenced encoder in term of power saving. For the "Akiyo" sequence, these methods achieve similar video quality performance. From the power saving perspective, the PDO has the best performance, followed by the FastME method. This is because in the PDO control method, the video encoder eventually only spends some computation on the talking women, saving a great amount of energy. In Figure 5.14 and Figure 5.15, the FastME and the PDO approaches have similar performance on power saving. However, the PDO approach achieves much better video quality, close to that of the ref video encoder, especially for fast motion sequence "Stefan". The FastME method reduces the power consumption at the expense of degraded video quality. From the results, we can see that the PDO video coding not only significantly reduces the power consumption, but also maintains the video quality for different motion sequences.

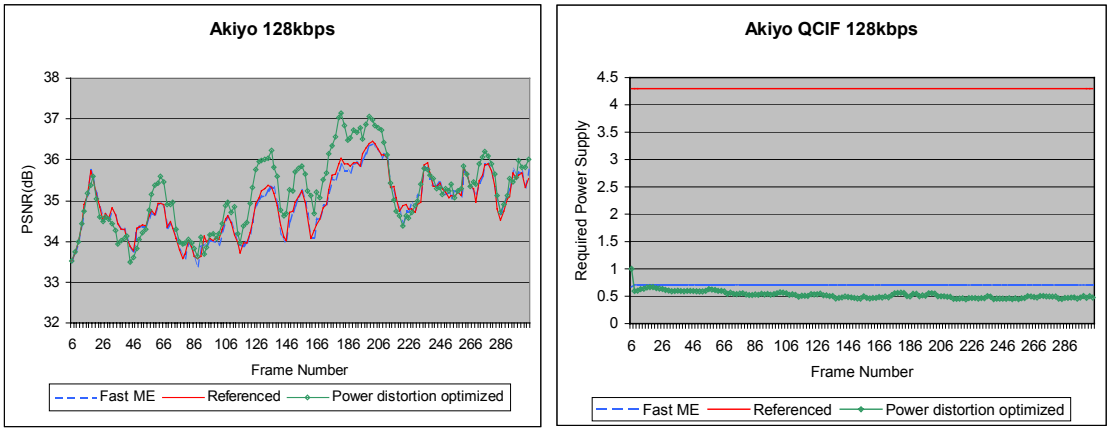


Figure 5.13 PSNR and power consumption comparison for “Akiyo” sequence.

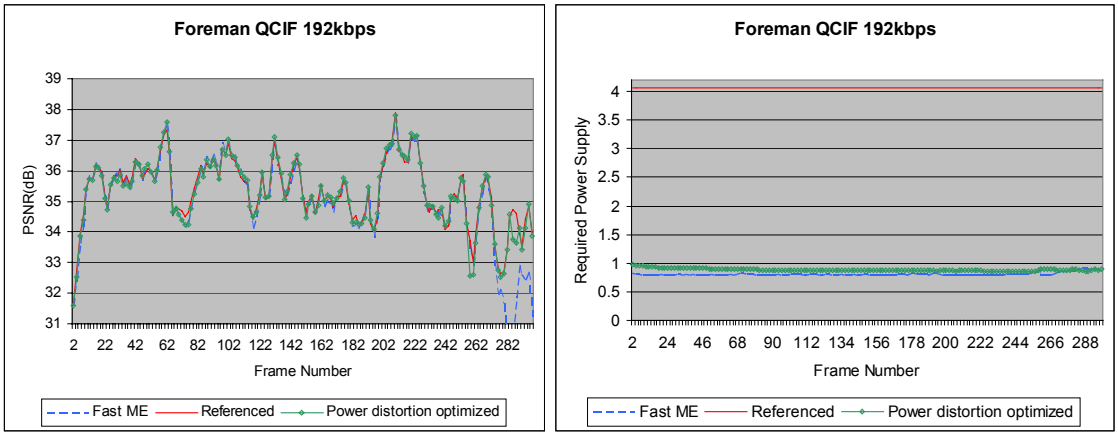


Figure 5.14 PSNR and power consumption comparison for “Foreman” sequence.

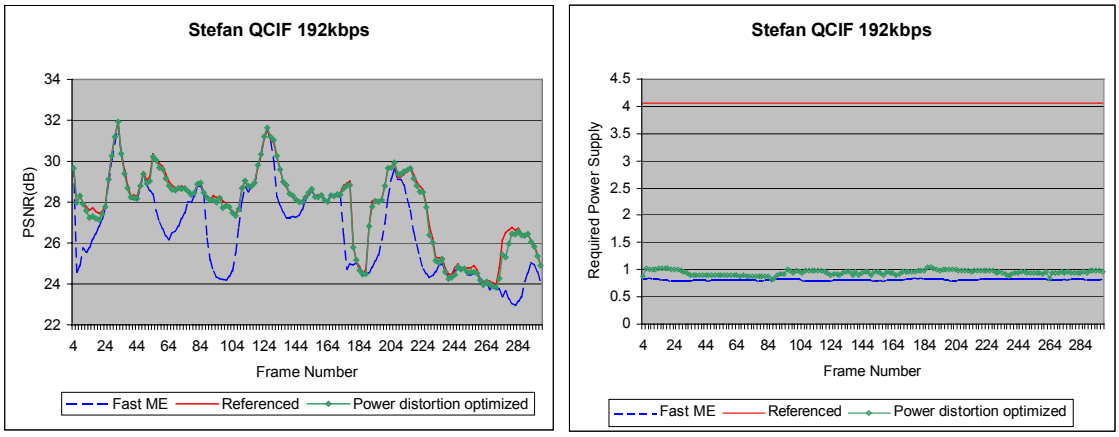


Figure 5.15 PSNR and power consumption comparison for “Stefan” sequence.

We further evaluate the performance of the proposed ideas using Simwattch, an instruction-set simulator enhanced with an instruction-level power model [10], to estimate the actual power consumption. Simwattch currently estimates the power consumption in a complete system simulation environment. It supports the analysis of power-efficient micro-architecture, application, compiler and operating system design decisions. In the experiments, we use the default setting assuming a 0.18-micron process technology at 600MHz. One of the advantages of using complexity scalable encoders is that they can achieve better performance on variable voltage processors, including Transmeta's Crusoe, AMD's K-6, TI OMAP processor family, and the IBM PowerPC 405LP. Such processors can operate at different clock frequencies and supply voltages, known as *dynamic voltage and frequency scaling* (DVS). In Simwattch, the dynamic power consumption of CMOS microprocessors is modeled as

$$P_d = CV_{dd}^2 \alpha F, \tag{5.23}$$

where C is the load capacitance, V_{dd} is the supply voltage, and F is the clock frequency. The activity factor, α, is a fraction between 0 and 1 indicating how often clock ticks lead to switching activity on average. Lowering the voltage results in a significant power reduction, since power is related to the square of the input voltage. In the experiments, we modify Simwattch and linearly relate V_{dd}, to the overall computation complexity in order to apply the DVS techniques.

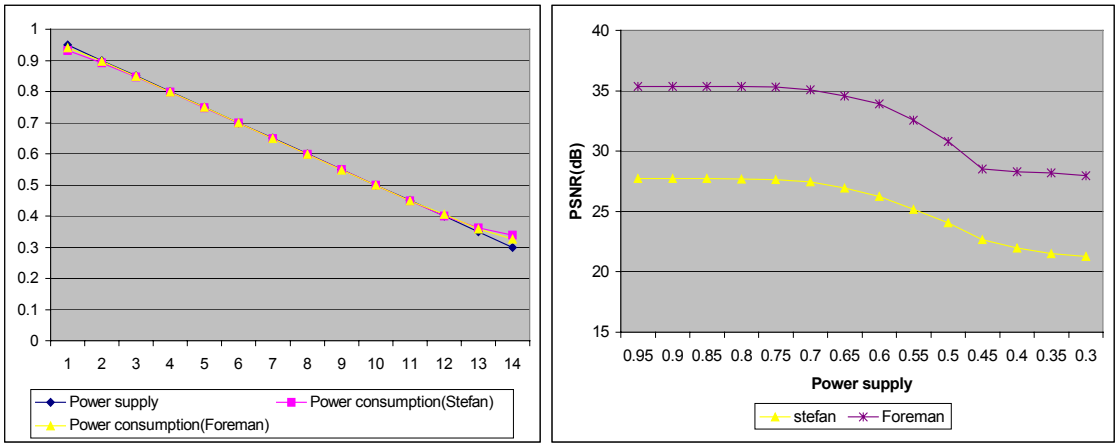
Table 5.2 presents the output of Simwattch for different test sequences using different approaches. The sequences are QCIF sequences coded at 15fps, 128kbps for 200 frames. The results verify that the PDO approach significantly reduces the power

consumption while preserving good video quality. When applying the DVS technique, we can save more power consumption.

Table 5.2 Power consumption (PC) comparison using Simwattch

Approach		akiyo	stefan	salesman	hall	foreman	carphone
Referenced	PC	$5.26x10^{11}$	$7.89 x10^{11}$	$6.11 x10^{11}$	$5.04 x10^{11}$	$6.91 x10^{11}$	$6.07 x10^{11}$
	PSNR(db)	44.09	26.75	39.57	39.87	33.99	36.93
Fast ME	PC	$2.21 x10^{11}$	$3.22 x10^{11}$	$2.35 x10^{11}$	$2.22 x10^{11}$	$3.02 x10^{11}$	$2.72 x10^{11}$
	PSNR(db)	44.10	25.66	39.56	39.89	33.91	36.90
PDO	PC	$2.20 x10^{11}$	$3.54 x10^{11}$	$2.49 x10^{11}$	$2.26 x10^{11}$	$3.36 x10^{11}$	$2.99 x10^{11}$
	PC with DVS	$1.30 x10^{11}$	$2.15 x10^{11}$	$1.31 x10^{11}$	$1.18 x10^{11}$	$2.33 x10^{11}$	$1.96 x10^{11}$
	PSNR(db)	44.13	26.53	39.60	39.90	33.95	36.88

Figure 5.16 shows the results of the PDO approach when the power supply changes. In (a), the x-axis is the time and the y-axis is the normalized power supply. One can observe that the proposed PDO approach is able to automatically adjust the system complexity according to the power supply level, which helps to extend the battery lifetime. As the power supply decreases, the video quality decreases gracefully. Note that for the referenced encoder and the FastME approach, they cannot adapt to the power supply level. Frames have to be dropped when the power supply is insufficient.



(a) (b)

Figure 5.16 PSNR and power consumption with insufficient power supply. (a) Power consumption, (b) Reconstructed video quality

5.6 Concluding Remarks

This chapter addresses the problem of how to efficiently manage the power consumption while preserving high video quality for video coding in futuristic ubiquitous environments. As opposed to the traditional video compression schemes, the power consumption and video distortion are considered jointly in this work. The work in this paper has two major parts. First, we show that the power-distortion optimized video coding can be considered as a multiple objective optimization problem. Similar to the rate-distortion model, we propose a R-C-D model that sufficiently describes the relationship between rate, complexity, and distortion. Second, valuable remarks on power-distortion optimization are provided and effective schemes are developed, which make us able to achieve good video quality while maximizing the battery service life. Being the first research effort, we believe the work here provides valuable insights into

the new power-distortion optimization research problem in futuristic ubiquitous environments.

CHAPTER 6

POWER-DISTORTION OPTIMIZED VIDEO CODING USING MOTION HISTORY INFORMATION

6.1 Motivation

In the upcoming pervasive computing era, wireless sensor networks are expected to rapidly emerge as a framework to carry out distributed and pervasive video applications such as environmental monitoring, battle field investigation, and video-surveillance, just to mention a few. In these environments, the small-size and low-cost tiny nodes, each equipped with a sensing device that collects information from the environment and a video coding device to transmit the information it through the network. Video cameras and monitors may be encountered in many places to observe ongoing activities and individuals. The vast accumulation of digital data requires new classes of video coding schemes to extend the battery operation time.

It is worth noticing that, unlike typical video applications on general platform, video applications in these scenarios are often characterized by low motion, in particular when no object is expected to move within the scene but in case of anomalies. When there is no motion, we can significantly slow down the coding procedure to preserve more power. The approach in Chapter 6 is shown to yield very good results but does no take this advantage into consideration, thus does not provide an optimal favorable energy/performance trade-off. In the context of video coding of a low-motion

scene, such as pervasive video surveillance, we expect to gain more power saving if the motion information of the coding video content can be achieved.

In this coding scenario, understanding the coding video content is an essential part to improve the performance. In this chapter, the goal is to develop schemes to capture the video content characteristics for power-distortion optimized video coding. Exploiting meta information with a broader scope can be potentially more effective, provided the overhead to obtain and maintain such information is low.

6.2 Hierarchical History of Motion Intensity

Similar to the way a video sequence is represented in block-based video coding, the motion intensity is determined hierarchically at block level, frame level, and sequence level, expressed by Ω_b, Ω_f, and Ω_s, respectively. We classify the intensity at each level to be in one of the following three distinct categories: *low*, *medium*, and *high* motion intensity. For simplicity, we only use the information from P frames to determine the motion intensity.

6.2.1 Block Level Motion Intensity:

The block level motion intensity (Ω_b) describes the intensity of motion correlation between current block and its neighboring left, top, and top-right blocks. Let Φ denote the local MV field that represents the set of MVs of the neighboring blocks. The smoother the local MV field, the higher motion correlation exists. In this case, the probability that the current block's MV has similar value as its neighboring blocks is

high. We define a parameter α to measure the smoothness of the local MV field as follows:

$$\alpha = \max\{|\, mv_x^{'} - \overline{mv_x}\,|, |\, mv_y^{'} - \overline{mv_y}\,|\}, (mv_x^{'}, mv_y^{'}) \in \Phi\,, \tag{6.1}$$

where $\overline{mv}$ is the average MV of Φ. Based on this measure, Ω_b is given by:

$$\Omega_b = \begin{cases} low & \alpha > L_2 \\ medium & L_1 < \alpha \le L_2 \\ high & \alpha \le L_1 \end{cases}, \tag{6.2}$$

where L_1 and L_2 are two block-level thresholds to be determined experimentally. In the experiments, we empirically set $L_1 = 1$ and $L_2 = 4$ for most video sequences.

6.2.2 Frame Level Motion Intensity:

The frame level motion intensity (Ω_f) describes the intensity of motion activity of one video frame. The intensity of motion activity of one block (ω) is classified into three categories:

$$\omega = \begin{cases} low & d = 0 \\ medium & 0 < d \le 4 \\ high & d > 4 \end{cases}, \tag{6.3}$$

where d is the magnitude of its associated MV (mv_x, mv_y), given by:

$$d = |mv_x| + |mv_y|. \tag{6.4}$$

The percentages of blocks with different values of ω are calculated as $p(\omega=low)$, $p(\omega=medium)$, and $p(\omega=high)$, respectively. For simplicity, we write $p(l)$, $p(m)$ and $p(h)$. Now, we calculate the average intensity of motion activity for all the blocks. Let X be a random variable on a finite integer set $\chi=\{x_1, x_2, x_3\}$, which

corresponds to the set of classification levels *{low, medium, high}*, with probability distribution function $p_k=P(X=x_k)$ that is given by the computed percentage distribution. We have $p_1=p(l)$, $p_2=p(m)$, and $p_3=p(h)$. The expected value of X is given by:

$$E(X) = \sum_{k=1}^{3} x_k p_k \ . \tag{6.5}$$

We define $\bar{\omega}$ as the expected intensity level, which is equal to low, medium, or high, depending upon the quantized value of *E(X)*. $\bar{\omega}$ can be used to estimate the intensity of motion activity of one frame, but may result in low estimation accuracy. Figure 6.1 and Figure 6.2 give two examples of the MV fields for video sequences "Akiyo" and "Stefan", at frame 37 and 91, respectively. The MVs were obtained by using full search ME with 16-pixel-size search region on 4x4 block. Figure 6.1 represents a typical head and shoulder scene with weak motion activity. Figure 6.2 contains fast-moving objects with strong motion activity. Notice that in Figure 6.1, the MV field is smooth, i.e., neighboring MVs do not differ a lot in direction and magnitude. Since most of the MVs are close to zero, a satisfactory result of low motion intensity at frame level is expected by averaging the intensities of block motion activities. In contrast, as seen in Figure 6.2, the MV field is more chaotic. The averaging may not reflect the presence of strong motion activity. We take this uncertainty into consideration when determining the frame level motion intensity.

Figure 6.1 Motion vector field for “Akiyo” at frame 37.

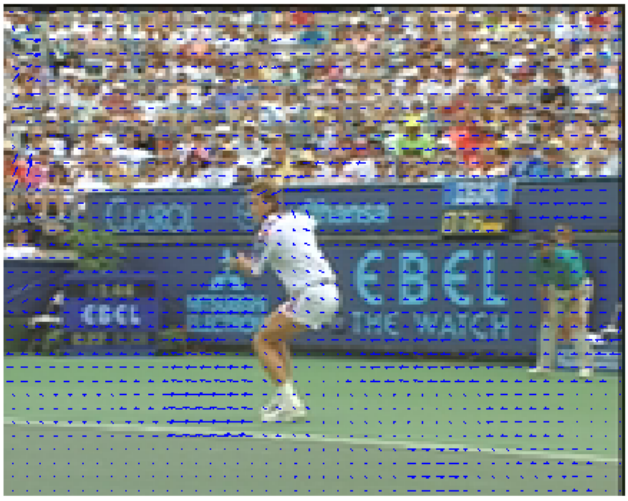


Figure 6.2 Motion vector field for “Stefan” at frame 91.

Given that the entropy is a measure of the uncertainty about the outcome of the random variable [27], we use the entropy to refine the classification of frame level motion intensity. The Shannon’s entropy *H(X)* of *X* is defined as:

$$H(X) = -\sum_{k=1}^{3} p_k \log_2 p_k . \tag{6.6}$$

Note that the entropy only depends on the probability distribution function of X. The frame level motion intensity is determined as following:

$$\Omega_s = \begin{cases} \overline{\omega} & H(X) \le TFL \\ medium & TFL \le H(X) \le TFH \\ high & H(X) > TFH \end{cases}, \tag{6.7}$$

where *TFL* and *TFH* are frame-level low and high thresholds. Experiments showed that TFL and TFH correspond to 0.35 and 0.6, respectively, for most video sequences.

6.2.3 Sequence Level Motion Intensity:

The sequence level motion intensity, denoted as Ω_s, represents the motion intensity of the whole sequence. Because the motion of one sequence tends to be similar continuously in time, the dominant frame level motion intensity can represent the motion characteristic of a sequence. Let MIH be the histogram of frame level motion intensity. $MIH = [p_0, p_1, p_2]$, where p_0, p_1, and p_2 are the percentages of frames with Ω_f equal to low, medium, and high intensity, respectively. Ω_s is determined by the dominant frame level motion intensity that has the maximum percentage in MIH. Table 6.1 provides the statistic data of the frame level motion intensities for various sequences with 200 frames. The last column shows the corresponding classified sequence level motion intensity. The results verify that descriptor Ω_s is efficient to describe the motion characteristics at sequence level.

Table 6.1 Sequence level motion intensity for typical sequences

Sequence	PERCENTAGE OF LOW MOTION INTENSITY FRAMES (%)	Percentage of medium motion intensity frames (%)	Percentage of high motion intensity frames (%)	Ω_s
Akiyo	0.838384	0.161616	0	low
Hall Monitor	0.79798	0.20202	0	low
Sean	0.747475	0.252525	0	low
Mother Daughter	0.151515	0.525253	0.323232	medium
Silent Voice	0.151515	0.666667	0.181818	medium
Table Tennis	0.050505	0.444444	0.505051	high
Stefan	0.010101	0.080808	0.909091	high
Mobile & Calendar	0	0.020202	0.979798	high

6.3 Using the Motion History Information

The hierarchical motion history structure enables better motion information retrieval of the coding video content, and therefore allows more aggressive complexity control. In this section, we will use this meta information to further save the computational power.

Remember that the block level motion intensity indicates the spatial motion correlation; the frame level motion intensity represents the average intensity of motion activity and the sequence level motion intensity shows the average motion intensity of the sequence. As a result, when the motion intensity at block level is high, and the motion intensities at frame and sequence level are both low, it indicates strong low motion present of the coding MB, and most probably it is a background MB. Therefore

we can apply more aggressive control and skip most of the time-consuming operation, after the dynamic complexity control is applied.

The proposed complexity control using motion history information works as follows:

Step 1: Initially, the motion intensity at each level is considered as low. The motion history image is initialized to 0.

Step 2: Get the frame level, and sequence level motion intensities for current MB.

Step 3: For current coding MB, compute block level motion intensity, if the motion intensity at block, frame, sequence level is high, low and low, we mark this MB to be a static or background low motion MB.

Step 4: During the greedy ME control, we only check the position with MV(0, 0). The SAD value is used later in INTRA / INTER coding selection.

Step 5: For AZMB determination of optimal mode selection in 5.21, we set *n* equal to 2 if the current MB is marked, otherwise *n* equal to 3 is used.

Step 6: Continue this procedure for the rest of MBs. Update the frame level and sequence level motion intensities accordingly. Here, we use the percentage MBs with low activity to determine the frame level intensity instead of using the entropy computation because of its simplicity.

In the proposed approach, the computational overhead introduced by the preprocessing of the hierarchical motion intensity framework includes two parts: update of the motion history image and computation of the motion intensities. The first part

only contains one addition and one comparison operation for each block. The second part builds up from the computation of block level motion intensity, which involves at most 6 subtractions, 3 additions, 6 absolute-value computations (for 3 neighboring MVs) and 1 comparison for a block. Compared with one SAD computation that requires N subtractions, N absolute values, and (N - 1) additions, where N is number of pixels in the block, this computation introduces low additional complexity. Remember that the computations of frame and sequence level motion intensities are performed only once for each frame and it mostly involves simple addition (for calculation of both the percentage distribution and the histogram) and comparison operations (for intensity classification). Overall, the preprocessing overhead on the hierarchical framework in the complexity control using motion history information is small and can be ignored.

6.4 Experimental Results

In this section, we evaluate the performance of the power-distortion optimized vide coding combined with the proposed hierarchical motion history structure. The power-distortion optimized H.263+ encoder is tested in the experiments. We compare the performance of video quality in term of y-PSNR value, power consumption in term of complexity, and output coding bit rate. The setup of the simulation is similar to the environments described in CHAPTER 5 Section 5.5 . The experiments are targeting the pervasive computing applications that will run for a long time, such as pervasive video surveillance on tiny sensors. We choose sequence “Hall_Moniotor” in the experiments since it is a typical sequence with content of video monitoring. We simulate one-hour

operation time in the experiments. Within the first 25 minutes, the video content is inactive and the camera captures the hall background with noise. In the next 15 minutes, two people are working in the hall and then leave, followed by another 10 minutes inactive status. The first 10 frames of "Hall_Monitor" sequence are used as the inactive content and the whole sequence is used as the active content. The experiments are perform individually for Full Search, Fast ME, Complexity control without using motion history information (CC only), and Complexity control using motion history information (CC with MHI) at 15fps, coding bit rate 384kbps and 64kbps, respectively.

Figure 6.3 and Figure 6.4 shows the luminance video quality at 384kbps and 64kbps respectively. We can observe that the dynamic complexity control with/without using the motion history information outperforms both the fast ME approach and the full search approach. This is because these two schemes can allocate the SAD computations more accurately to the areas where it is needed and thus improve the overall video quality. It is worth to note that although the full search approach can find the minimum distortion, but with a much more chaos motion vector field, which affects the compression efficiency, especially in high coding bit rates, shown in Figure 6.3. On the other hand, the approach of complexity control using motion history information has video quality close to complexity control without using the motion history information. It is even better within the inactive period at low bit rates 64kbps, since the computational power is more intelligently distributed according to the motion history information.

Figure 6.5 shows the complexity comparisons. We can notice that the proposed complexity control with/without using the motion history information outperforms the fast ME approach. (It is easy to understand that it has the highest complexity. For simplify purpose, the complexity of full search is not shown here.) One can also easily observe that the proposed dynamic complexity control when combined with exploiting the motion history information significantly reduce the video coding complexity, compared to not using it. Results concerning the number of bits used are shown in Figure 6.6 and Figure 6.7. The results verify that the proposed complexity control with/without using the motion history information does not affect the robustness of the rate control algorithm. It is interesting to see that using the motion history information produces a little less bit rate than other approaches.

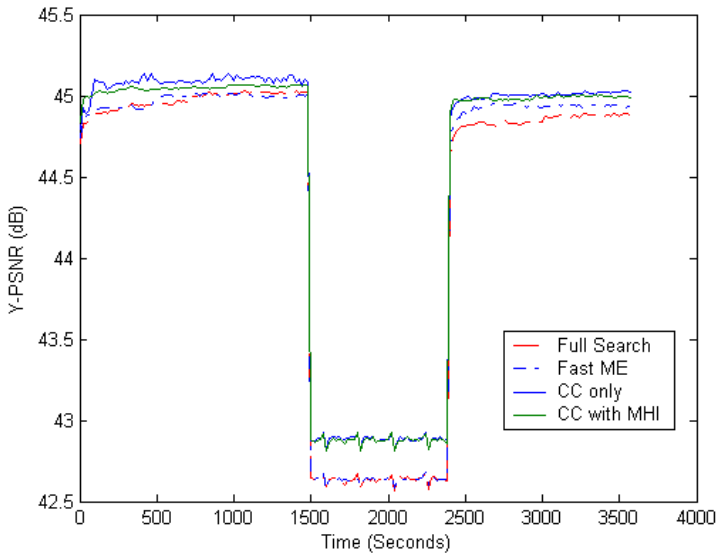


Figure 6.3 Y-PSNR comparisons, coding bit rate at 384kbps, 15fps.

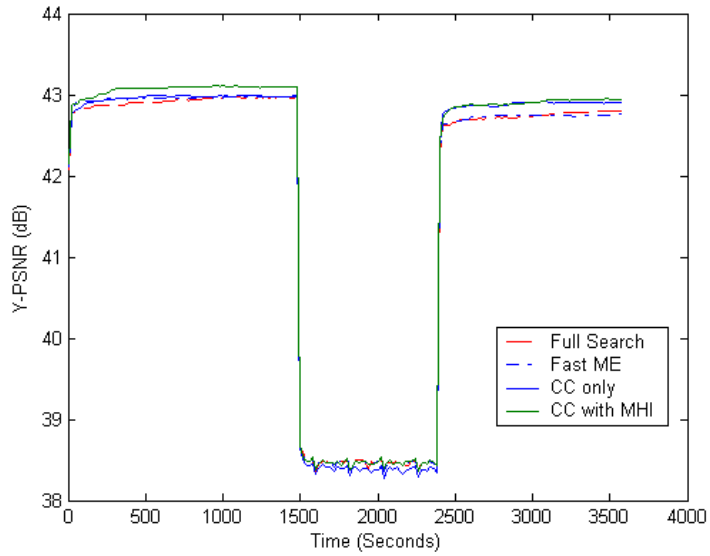


Figure 6.4 Y-PSNR comparisons, coding bit rate at 64kbps, 15fps.

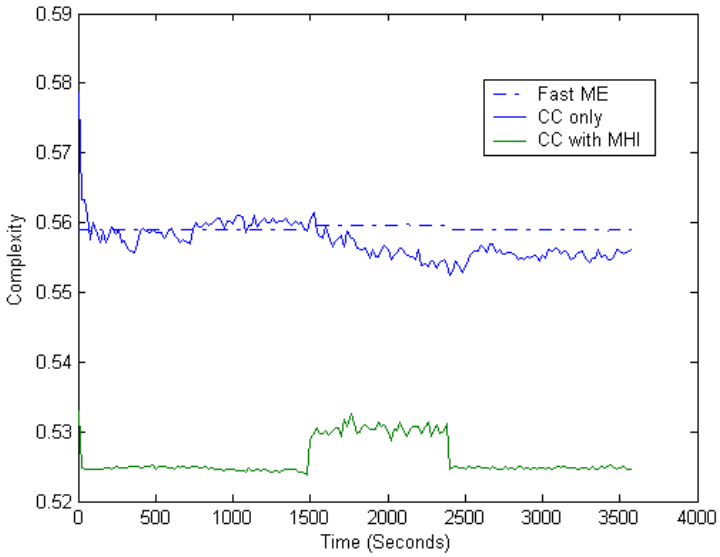


Figure 6.5 Complexity comparisons, coding bit rate at 384kbps, 15fps.

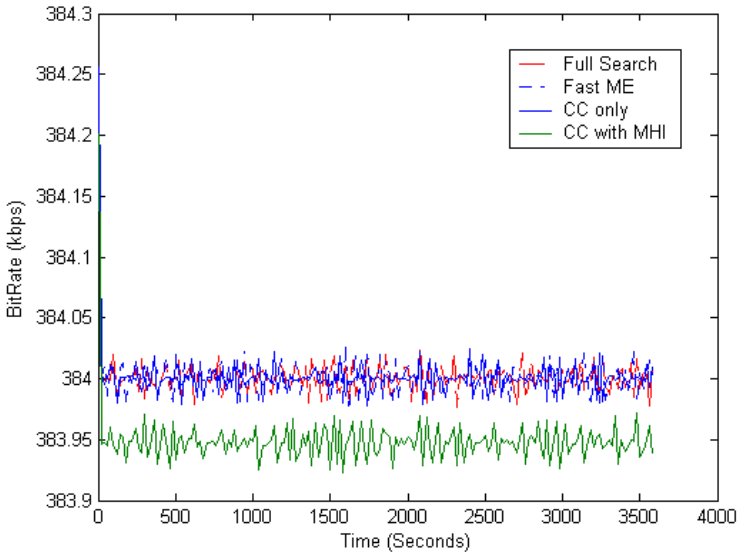


Figure 6.6 Output coding bit rates, target coding bit rate 384kbps.

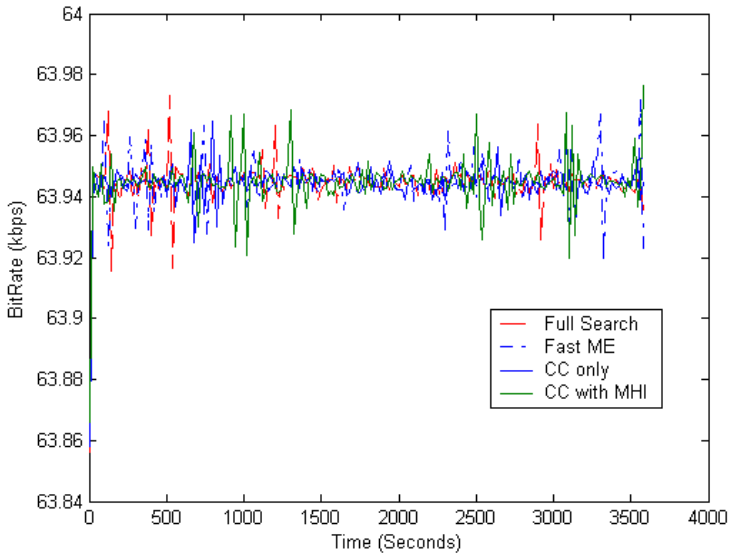


Figure 6.7 Output coding bit rates, target coding bit rate 64kbps.

6.5 Concluding Remarks

The proposed hierarchical motion history structure framework effectively captures, retrieves and exploits the subtle motion information of a video sequence. This motion history information provides a better understanding of the video content the video encoder is encoding. It can be used in fast motion estimation, video segmentation, video analysis, etc. In this chapter, the motion history hierarchy is combined with the dynamic complexity control for power-distortion optimized video coding in order to gain more power saving. The experimental results certify that the meta information contained in the hierarchy can be very useful in improving the power saving performance for persuasive computing applications.

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cyirci,, "Wireless Sensor Networks: A Survey," *Computer Networks*, 38(4):393-422, 2002.

[2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communication Magazine*, pp. 102 - 114, August 2002.

[3] P. Agrawal, J. Chen, S. Kishore, P. Ramanathan, and K. Sivalingam, "Battery Power Sensitive Video Processing in Wireless Networks," *Ninth IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, vol. 1, 8-11 Sept. 1998, pp.116 – 120.

[4] S. M. Akramullah, I. Ahmad and M. L. Liou, "A Data-Parallel Approach for Real-Time MPEG-2 Video Encoding," *Journal of Parallel and Distributed Computing*, Vol. 30, No. 2, pp. 129-146, Nov. 1995.

[5] S. M. Akramullah, I. Ahmad, and M. L. Liou, "Optimization of H.263 Video Encoding using a Single Processor computer: Performance Tradeoffs and Benchmarking," *IEEE Transactions On Circuits and System for Video Technology,*vol. 11, pp. 901 - 915, August 2001.

[6] R. T. Apteker, J. A. Fisher, V. S. Kisimov, and H. Neishlos "Video Acceptability and Frame Rate," *IEEE Multimedia*, vol 2, No. 3, pp. 32-40, 1995.

[7] N. August, D. Ha, "On the Low-Power Design of DCT and IDCT for Low Bit-rate Video Codecs", *14th Annual IEEE International ASIC/SOC Conference*, 12-15, Sept. 2001.

[8] L. Benini, A. Bogliolo and G. De Micheli, "A Survey of Design Techniques for System-Level Dynamic Power Management," *IEEE Transactions on very Large Scale Integration (VLSI) Systems*, vol. 8, pp. 299-316, June 2000.

[9] T. Berger, Rate Distortion Theory, Prentice Hall, Englewood Cliffs, NJ, 1984.

[10] D. Brooks, V. Tiwari and M. Martonosi, "Wattch: A Framework for Architectural-Level Power Analysis and Optimizations", in *Proc. 27th Ann. Int. Symp. Computer Architecture*, 2000, pp. 83-94.

[11] L. Bowen, R. Zarr and S. Denton, "A Microcontroller Controlled Battery Fuel Gauge and Charger," in *Proceedings of the Ninth Annual Battery Conference on Applications and Advances*, pp. 11-13, 1994.

[12] W. Burleson, P. Jain, and S. Venkatraman, "Dynamically Parameterized Architectures for Power-aware Video Coding: Motion Estimation and DCT," *Second International Workshop on Digital and Computational Video*, 8-9 Feb. 2001.

[13] J. Canaris, "A VLSI Architecture for the Real Time Computation of Discrete Trigonometric Transforms," *J. VLSI Signal Processing*, vol. 5, pp. 95-104, 1993.

[14] P. Caumont, L. Moigne, C. Rombaut, X. Muneret and P. Lenain, "Energy Gauge for Lead-Acid Batteries in Electric Vehicles," *IEEE Transactions on Energy Conversion*, vol. 15, Sept. 2000.

[15] J. R. Corbera and S. Lei, "Rate Control in DCT Video Coding for Low-delay Communications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 172–185, Feb. 1999.

[16] N. Chaddha and M. Vishwanath, "A Low Power Video Encoder with Power, Memory and Bandwidth Scalability," *Ninth International Conference on VLSI Design*, 3-6 Jan. 1996, pp.358 – 363.

[17] S. Chakraborty and D. K. Y. Yau, "Predicting Energy Consumption of MPEG Video Playback on Handhelds," *IEEE International Conference on Multimedia and Expo*," vol. 1, 26-29 Aug. 2002, pp.317 – 320.

[18] Y. S. Chan and J. W. Modestino, "A Joint Source Coding-power Control Approach for Video Transmission over CDMA Networks," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 10, Dec. 2003, pp.1516 – 1525.

[19] Y. Chang and C. Wang, "New Systolic Array Implementation of the 2D Discrete Cosine Transform and Its Inverse," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 5, pp. 150-157, 1995

[20] J. Chen and K. J. R. Liu, " Low-power Architectures for Compressed Domain Video Coding Co-processor," *IEEE Transactions on Multimedia*, vol. 2, no. 2, June 2000, pp.111 – 128.

[21] C. K. Cheung and L. M. Po, "A Hierarchical Block Motion Estimation Algorithm Using Partial Distortion Measure," in *Proceeding of ICIP'97*, 1997, vol. 3, pp. 606-609.

[22] T. Chiang, Y. Q. Zhang, "A New Rate Control Scheme Using Quadratic Rate Distortion Model," *IEEE Transactions on Circuits and Systems for Video Technology*, vol.7, pp. 246 – 250, Feb. 1997.

[23] S. Chien, S. Ma, and L. Chen, "Efficient Moving Object Segmentation Algorithm Using Background Registration Technique," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 7, pp. 577-586, July 2002.

[24] N. Cho and S. Lee, "Fast Algorithm and Implementation of 2D Discrete Cosine Transform," *IEEE Trans. Circuits and Systems*, vol. 38, pp. 297-305, 1991.

[25] N. I. Cho and S.U. Lee, "DCT Algorithms for VLSI Parallel Implementation," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 38, pp. 121-127, 1990.

[26] J. R. Corbera and S. Lei, "Rate Control in DCT Video Coding for Low Delay Communications," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 9, pp. 172 – 185, Feb. 1999.

[27] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley Series in Telecommunications, Wiley, New York, 1991.

[28] K. Deb, Multi-Objective Optimization using Evolutionary Algorithms, Wiley, 2001.

[29] W. Ding and B. Liu, "Rate Control of MPEG Video Coding and Recording by Rate–quantization Modeling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 12–20, Feb. 1996.

[30] V. L. Do and K. Y. Yun, "A Low-power VLSI Architecture for Full-search Block-matching Motion Estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, Issue 4, Aug. 1998 pp. 393 – 398.

[31] P. Duhamel and C. Guillemot, "Polynomial Transform Computation of the 2D DCT," *Proc. Int'l Conf. Acoustics, Speech, and Signal Processing*, pp. 1,515-1,518, 1990.

[32] T. Ebrahimi and F. Pereira, "The MPEG-4 book", ISBN: 0-130-61621-4, Prentice Hall PTR, July 2002.

[33] Y. Eisenberg, T. N. Pappas, R. Berry, and A. K. Katsaggelos, "Minimizing Transmission Energy in Wireless Video Communications," *International Conference on Image Processing*, vol. 1, 7-10, Oct. 2001, pp.958 – 961.

[34] M. A. Elgamel and M. A. Bayoumi, "A Comparative Analysis for Low Power Motion Estimation VLSI Architectures," *IEEE Workshop on Signal Processing Systems*, *SiPS 2000*, 11-13 Oct. 2000 pp.149 – 158.

[35] M. A. Elgamel, A. M. Shams, X. Xi, and M. A. Bayoumi, "Enhanced Low Power Motion Estimation VLSI Architectures for Video Compression", *The 2001 IEEE International Symposium on Circuits and Systems, 2001. ISCAS 2001*, vol. 4, 6-9 May 2001, pp. 474 – 477.

[36] E. Feig and S. Winograd, "Fast Algorithms for the Discrete Cosine Transform," *IEEE Trans. Signal Processing*, vol. 40, pp. 2,174-2,193, 1992.

[37] B. Furht, J. Greenberg, and R. Westwater, Motion Estimation Algorithms for Video Compression, Springer, 1 edition, Nov. 1996.

[38] K. Gaedke, H. Jeschke, and P. Pirsch, “VLSI Based MIMD Architecture of a Multiprocessor System for Real-time Video Processing Applications,” *Journal of VLSI Signal Processing,* vol. 5, no. 2/3, pp. 159-169, 1993.

[39] M. J. Gormish, Source Coding with Channel, Distortion and Complexity Constraints, Ph.D. thesis, Stanford University, Mar. 1994.

[40] V. K. Goyal and M. Vetterli, "Computation-Distortion Characteristics of Block Transform Coding," *Proc. IEEE Int. Conf. Acoust., Speech, and Signal ICASSP'1997*, vol. 4, pp. 2729-2732, Munich, Germany, Apr. 1997..

[41] H. M. Hang and J. J. Chen, “Source Model for Transform Video Coder and Its Application. I. Fundamental Theory,” *IEEE Trans. CSVT*, vol.7, Apr. 1997.

[42] Y. He, I. Ahmad, and M. L. Liou, “A Software-based MPEG-4 Video Encoder Using Parallel Processing,” *IEEE Transactions on Circuits and Systems for Video Technology*, Special issue on Image and Video Processing for Emerging Interactive Multimedia Services, September 1998.

[43] Z. He, Y. Liang, L. Chen, I. Ahmad, and D. Wu, "Power-Rate-Distortion Analysis for Wireless Video Communication under Energy Constraint," *IEEE Transactions on Circuits and System for Video Technology*, vol. 14, no. 5, pp. 645-658, May 2005.

[44] Z. He, C. Tsui, K. Chan, and M. Liou, “Low-power VLSI Design for Motion Estimation Using Adaptive Pixel Truncation,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, Issue 5, Aug. 2000 pp. 669 – 678.

[45] Z. He and S. K. Mitra, “A Unified Rate-Distortion Analysis Framework for Transform Coding,” *IEEE Transactions on Circuits and System on Video Technology*, vol. 11, pp. 1221 -1236, Dec. 2001.

[46] Z. He and S. K. Mitra, "A Linear Source Model and a Unified Rate Control Algorithm for DCT Video Coding," *IEEE Transactions on Circuits and System on Video Technology*, vol. 12, pp. 970 - 982, Nov. 2002.

[47] Intel, "Using Streaming SIMD Extensions in a Fast DCT Algorithm for MPEG Encoding," *Intel Application Note AP-817*, Order No: 243651-002.

[48] Intel, "Using Streaming SIMD Extensions in a Motion Estimation Algorithm for MPEG Encoding," *Intel Application Note AP-818*, Order No: 243652-002.

[49] I. Ismaeil, A. Docef, F. Kossentini, and R. Ward, "Computation-performance Control for DCT-based Video Coding," *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2000. ICASSP '00. Proceedings*. 2000, vol. 6, 5-9 June 2000.

[50] http://www.cc.gatech.edu/fce/house/

[51] ISO/IEC CD 11172/2 (MPEG-1 Video), Information technology----Coding of Moving Pictures and Associated Audio for Digital Storage Media at Up to About 1.5 Mbits/s: Video, 1993.

[52] ISO/IEC CD 13818/2 (MPEG-2 Video), Information technology----Generic Coding of Moving Pictures and Associated Audio Information: Video, 1995.

[53] ISO/IEC CD 14496/2 (MPEG-4 Video), Information technology----Coding of Natural/Visual Objects, Part 2: Visual, 1999.

[54] ISO/IEC, Information Technology – Generic Coding of Audio Visual Object, ISO/IEC JTC 1/SC 29/WG 11 N 2502, Atlantic City, 1998.

[55] ITU-T, Video Codec for Audiovisual Services at p x 64 kbits/s, Recommendation H.261, Geneva, 1990.

[56] ITU-T, Video Codec for Low Bit-Rate Communication, Recommendation H.263, Dec. 1995.

[57] ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC, Advance Video Coding, Draft, Oct. 2002.

[58] Z. Ji, Q. Zhang, W. Zhu, and Y. Zhang, "End-to-end Power-optimized Video Communication over Wireless Channels," *IEEE Fourth Workshop on Multimedia Signal Processing*, 3-5 Oct. 2001, pp.447 – 452.

[59] I. Kim and H. M. Kim, "Transmit Power Optimization for Video Transmission over Slowly-varying Rayleigh-fading Channels in CDMA Systems," *IEEE Transactions on Wireless Communications*, vol. 3, no. 5, Sept. 2004, pp.1411 – 1415.

[60] I. M. Kim, H. M. Kim, and D. G. Sachs, "Power-distortion Optimized Mode Selection for Transmission of VBR Videos in CDMA Systems," *IEEE Transactions on Communications*, vol. 51, no. 4, Apr. 2003, pp.525 – 529.

[61] P. Kuhn, Algorithms, Complexity Analysis and VLSI Architectures for MPEG-4 Motion Estimation, Kluwer Academic Publishers, Boston, 1999.

[62] P. Kumar and M. Srivastava, "Power-aware Multimedia Systems Using Run-time Prediction," *Fourteenth International Conference on VLSI Design*, Jan. 2001 pp.64 – 69.

[63] T. Lan and A. H. Tew, "Power Optimized Mode Selection for H.263 Video Coding and Wireless Communications," in *Proceedings of International Conference on Image Processing*, 1998, vol. 2, pp. 113-117.

[64] P.Z. Lee and F.Y. Huang, "An Efficient Prime-Factor Decomposition of the Discrete Cosine Transform and Its Hardware Realization," *Proc. Int'l Conf. Acoustics, Speech, and Signal Processing*, pp. 20.5.1-20.5.4, 1985.

[65] Y. Liang, Z. He and I. Ahmad, "Analysis and Design of Power Constrained Video Encoder," in *IEEE 6th CAS Symposium on Emerging Technologies: Frontiers of Mobile and Wireless Communication (MWC'04)*, 2004.

[66] J. Lin and A. Ortega, “Bit-rate Control Using Piecewise Approximation Rate-Distortion Characteristics,” *IEEE Trans. CSVT*, vol. 8, Aug. 1998.

[67] X. Lu, Y. Wang, and E. Erkip, “Power Efficient H.263 Video Transmission over Wireless Channels,” *Proceedings of 2002 International Conference on Image Processing*, Rochester, New York, Sept. 2002.

[68] X. Lu, E. Erkip, Y. Wang, and D. Goodman, “Power Efficient Multimedia Communication over Wireless Channels,” *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 10, Dec. 2003, pp.1738 – 1751.

[69] X. Lu, T. Fernaine, and Y. Wang, “Modeling Power Consumption of a H.263 Video Encoder,” *Proceedings of the 2004 International Symposium on Circuits and Systems ISCAS '04*, vol. 2, 23-26, pp. 77-80, May 2004.

[70] K. Nguyen-Phi, A. Docef, and F. Kossentini, “Quantized Discrete Cosine Transform: A combination of DCT and scalar quantization,” in *Proc. ICASSP*, pp.3197-3200, 1999.

[71] A. V. Oppenheim and R.W. Shafer, Digital Signal Processing, Englewood Cliffs, N.J.: Prentice Hall, 1975.

[72] I.M. Pao and M. Sun, "Modeling DCT Coefficients for Fast Video Encoding," *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 9, pp. 608-616, June 1999.

[73] N. Paragios and G. Tziritas, “Detection and Location of Moving Objects Using Deterministic Relaxation Algorithms,” in *Proceedings of IEEE International Conf. Pattern Recognition*, 1996.

[74] L.D. Paulson, “Will Fuel Cells Replace Batteries in Mobile Devices,” *IEEE Computer*, Nov. 2003, pp. 10-12.

[75] P. Pirsch, N. Demassieux, and W. Gehrke, “VLSI Architectures for Video Compression - A Survey”, *Proceedings of the IEEE*, vol. 83, Feb. 1995, pp. 220-246.

[76] R. Rao, S. Vrudhula, D.N. Rakhmatov, "Battery Modeling for Energy Aware System Design, " *IEEE Computer*, Volume: 36, Issue: 12 , pp.77 – 87, Dec. 2003.

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

[78] V. Rodriguez, "Optimal Coding Rate and Power Allocation for the Streaming of Scalably Encoded Video over a Wireless Link," *IEEE ICASSP*, May 2004.

[79] M. Schaar and Y. Andreopoulos, "Rate-distortion-complexity Modeling for Network and Receiver Aware Adaptation," *IEEE Transactions on Multimedia*, vol. 7, no. 3, pp. 471-479, June 2005.

[80] K. Shen, G. W. Cook, L. H. Jamieson, and E. J. Delp, "An Overview of Parallel Processing Approaches to Image Compression," *SPIE Conf. Image and Video Compression,* vol. 2186, Feb. 1994, pp. 197--208.

[81] T. Sikora, "The MPEG-4 Video Standard Verification Model," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 7, pp. 19–31, Feb. 1997.

[82] K. Stuhlmuller, N. Farber, M. Link, and B. Girod, "Analysis of Video Transmission over Lossy Channels," *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 1012~1032, June 2000.

[83] X. Tian, "Efficient Transmission Power Allocation for Wireless Video Communications," *IEEE Wireless Communications and Networking Conference, WCNC. 2004*, vol. 4, 21-25 Mar. 2004, pp. 2058 – 2063.

[84] Z. Xiong, S. S. Cheng, and M. Fossorier, "Joint Source-channel Video Coding for a Power Constrained Noisy Channel," *Conference Record of the Thirty-Third Asilomar Conference on Signals, Systems, and Computers*, vol. 1, 24-27 Oct. 1999 pp.317 – 320.

[85] T. Yamada, M. Ikekawa, and I. Kuroda, "Fast and Accurate Motion Estimation Algorithm By Adaptive Search Range and Shape Selection," in *Proc. of ICASSP'2005*, Philadelphia, USA, Mar. 2005.

[86] F. Zhai, Y. Eisenberg, C. E. Luna, T. N. Pappas, R. Berry, and A. K. Katsaggelos, "Joint Source-channel Coding and Power Allocation for Energy Efficient Wireless Video Communications," *Proc. 41st Allerton Conf. Communication, Control, and Computing*, Oct. 2003.

[87] Q. Zhang, Z. Ji, W. Zhu, and Y. Zhang, "Power-minimized Bit Allocation for Video Communication over Wireless Channels," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 6, June 2002, pp.398 – 410.

[88] Q. Zhang, W. Zhu, Z. Ji, and Y. Zhang , "A Power-Optimized Joint Source Channel Coding for Scalable Video Streaming over Wireless Channel," *IEEE International Symposium on Circuits and Systems*, vol. 5, 6-9 May 2001, pp.137 – 140.

[89] X. J. Zhao, Y. W. He, S. Q. Yang, and Y. Z. Zhong, "Rate Allocation of Equal Image Quality for MPEG-4 FGS Video Streaming," *Packet Video Workshop*, Apr. 2002.

[90] D. W. Zheng, I. Ahmad, and M. Liou, "Real-Time Software Based MPEG-4 Video Encoding," *2nd IEEE MPEG-4 Workshop and Exhibition*, San Jose, CA, June 2001.

BIOGRAPHICAL INFORMATION

Yongfang Liang received the B.S. and M.S. degrees from Zhongshan(Sun Yat-sen) University, Guangzhou, China, in 1998 and 2001 respectively, in Electrical Engineering and Communication and Information System, respectively. From 2001 to 2002, he was with HuaWei Technologies Corporation, Shenzhen, as a Researcher and Software Engineer in the Multimedia Department, working on the video codec of a video conferencing set top box. In Summer 2004, he was with Nokia Research Center, Irving, TX, as an intern in the Video Group. He was responsible to develop transcoding schemes for a video splicing application on Nokia's cell phones. In Summer 2005, he was with Sun Microsystems Laboratories, Menlo Park, CA, as an intern in the digital rights management group. He was responsible to develop conditional access system for Sun's video streaming project. He also participated in Sun's Open Media Commons activities. He pursued his Ph.D. degree in Computer Science and Engineering, at the University of Texas at Arlington. At the mean time, he actively participated in various projects in the multimedia lab and the COPSE project of the IRIS (Institute for Research In Security) lab at UTA. His research interests include video compression, wireless multimedia communication, and image processing. He is a member of the IEEE and SPIE.