Resource Constrained Video Coding/Adaptation

Yong Wang

Submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

in the Graduate School of Arts and Sciences

Columbia University

2005

To my parents, who give me the meaning of life

To my wife, who gives me the meaning of love

ABSTRACT

Resource Constrained Video Coding/Adaptation

Yong Wang

Typical video coding systems focus on optimization of the video quality subject to certain resource constraints, such as bandwidth. The rate distortion optimization framework has been widely used in optimizing the performance tradeoffs. One fundamental issue involved is the difficulty in estimating the rate-distortion relationships under different design choices. Likewise in a video adaptation system, where encoded videos are adapted to meet dynamic resource requirements, the relationships between resources and video quality are difficult to estimate analytically. In this thesis, we extend the rate-distortion concepts to a broader framework based on utility function (UF), which models the relations among resources, video utility, and adaptation operations. We then propose a novel content-based paradigm to automatically predict the utility function of a video and determine the optimal video adaptation operation. Our approach is flexible in that different types of utilities, resources, or operations can be easily incorporated. Our content-based prediction approach is shown to be accurate. In the second half of the thesis, we further extend the framework to investigate the influence of power resource constraint. We develop a joint power-rate-distortion optimization method to achieve optimal video quality under joint constraints of available bandwidth and power.

Video adaptation modifies an existing encoded video stream to different frame rates, spatial resolutions, or others forms to meet dynamically changing conditions of networks or client platforms. Multiple dimensional adaptation (MDA) can be constructed by combining more than one type of adaptation operations. Neverthe-

less, selection of the best MDA operation among various choices is often done in an ad hoc way and hard to generalize. To provide a systematic solution, we present a framework based on a general utility function, which models the relationships among video quality, resources, and adaptation operations. We show utility functions are correlated to the content features of the video, which can be automatically extracted in real time. Machine learning methods are then developed to train systems which predicts the best adaptation operation according to the content features. We evaluate the proposed methods over videos compressed by MPEG-4 and motion compensated embedded zeroblock coding (MC-EZBC) respectively. To consider different quality measurements, perceptual subjective evaluation is also conducted, in addition to objective quality measurements. The characteristics of MDA over these codecs are analyzed, the utility functions are constructed, and the performance of content based operation prediction is validated with excellent performance.

Complexity aware video coding is important for mobile/wireless devices where the computational capability or power resource is restricted. Emerging video coding standards like H.264 achieve significant improvements in video quality, at the expense of greatly increased computational complexity at both the encoder and the decoder. We propose a systematic solution for complexity optimized video decoding by extending the conventional rate-distortion framework. A complexity-control algorithm is also developed to meet the overall specified target complexity level and keep the complexity consistent significant complexity reduction (up to 60% in motion compensation) with little quality degradation (less than 0.3dB).

Contents

List of Figures

List of Tables

Acknowledgements

I would like to deliver my sincere gratitude to Prof. Shih-Fu Chang who supervised my Ph.D. research work at Columbia University. Prof. Chang always keeps vivid view on my research work and provides me critical direction or inspiration on each important stage, at the same time leaving me enough freedom to explore the problems. The knowledge I learn from Prof. Chang is not only embodied in this thesis, but also the training about rigorous scholarship and solid research, which I will continue benefiting from.

I would like to thank Prof. Mihaela van der Schaar from University of California at Los Angeles, who co-advised part of my thesis work, Dr. Alexander C. Loui from Kodak, who collaborated with me for several projects, Dr. Jae-Gon Kim from ETRI, a hard-working and help-willing guy with whom I had pleasant collaboration during his visiting stay at Columbia. I also learn from the discussion with Dr. Alexis M. Tourapis from DoCoMo, Dr. Deepak S. Turaga and Dr. Ching-Yung Lin from IBM. Dr. Daniel Tretter and Dr. Tong Zhang provided me an opportunity to work as a summer intern at HP labs so that I can extend my research scope.

I thank my thesis committee members: Prof. Dan Rubenstein, Prof. Xiaodong Wang, Prof. Jason Nieh, and Dr. Anthony Vetro, for their kind work during my defense and the valuable comments they provide on my thesis.

During the course of this thesis work I have learned from interacting with my colleague in DVMM group: Winston Hsu, Jessie Hsu, Lyndon Kennedy, Barry A. Rafkind, Eric Zavesky, Alejandro Jaimes, Ana Belen Benitez and Shahram Ebadollahi. Special thanks to Tian-Tsong Ng and Gounyoung Kim for their work on MC-EZBC project, Lexing Xie and Prof. Hari Sundaram for their work on VisGenie project, Dongqing Zhang for his patience with my Matlab and LaTex questions.

Also thank my friends at Columbia for the pleasant moments with them: Hairuo Liu, Lin Li, Chuxiang Li, Dong Guo, Kai Yang, Kai Li, Qi Duan, Ting Song and Zhipeng Sun.

Lastly but most importantly, I want to appreciate my family: my parents, my wife and my brother for their consecutive encouraging and unconditional support during my Ph.D. study. It is being with them that makes my Ph.D. life in New York fulfilled with love and happiness.

Chapter 1

Introduction

1.1 Background

We have witnessed the dramatic development of video coding technologies and applications since H.261. There are three primary driving forces that boost such development. First, the requirement by the new video applications greatly stimulate the demand of new video coding technology. Universal media access (UMA) [49] requires video content to be accessed in a ubiquitous manner. The prevalence of webcam, digital camera and camcorder makes it extremely convenient to generate video data. At the same time emerging applications such as HD-DVD, HDTV, and 3G multimedia services greatly raise the level of requirements for the video coding performance in both efficiency and quality. Second, the continuous efforts from academies and industries have resulted in exciting progress in theories, algorithms, and implementations of video coding, which have been successfully incorporated into the state-of-the-art video coding. Notable progresses include much more sophisticated algorithms in motion estimation, signal transform, adaptive entropy coding, and multi-dimensional scalable coding. Last, the progress in computer hardware following the Moore's rule make it possible to adopt sophisticated computations in video coding, giving support for the techniques that are computationally complex but critical for video quality enhancement.

The rate-distortion(R-D) theory is a widely used principle in designing video coding systems in order to achieve the optimal tradeoff between video quality and required bandwidth. Nevertheless, it is non-trivial to extend the R-D theory to accommodate complex requirement when various operation parameters, resource constraints, and quality requirements are considered. Such complex requirements arise when the video signals are delivered through heterogenous network into diverse end devices, which is the typical scenarios defined by UMA as illustrated in Figure 1.1. To address this problem there are two approaches. First, the conventional R-D framework is extended to explicitly model additional constraints such as power consumption. Alternatively, an adaptation system can be introduced as an intermediary, which takes encoded videos as input and adapt the video into a new one conforming with the new resource constraints (power, display, bandwidth etc). We call these approaches as resource constrained video coding/adaptation.

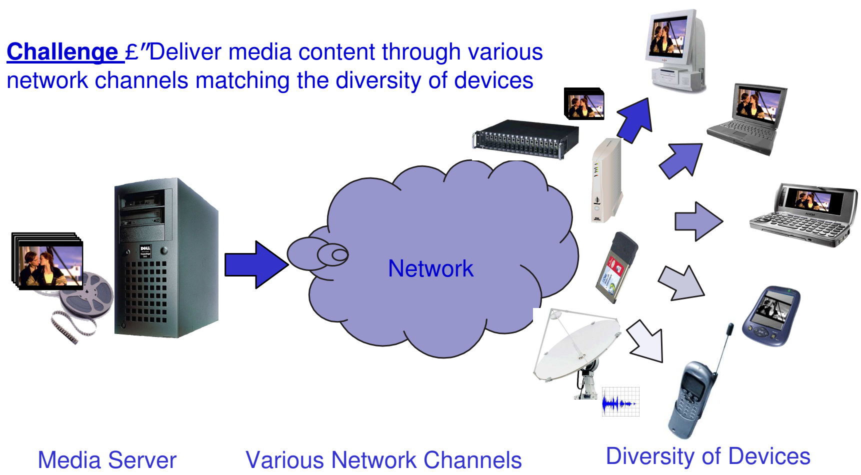


Figure 1.1: Universal media access (UMA) involves the delivery of media content through heterogeneous networks to diverse client platforms.

In this thesis we address several open issues related to resource constrained video coding/adaptation. Specifically we solve two problems: first, automatic selection of optimal operations available in a multiple dimensional space; second, extension of the R-D optimization framework to incorporate the power consumption constraint. In this chapter we first provide a brief overview of the state-of-the-art research on video coding/adaptation (Section 1.2). The open issues and the motivation of the thesis work are further described (Section 1.3). The contribution and organization of the thesis are summarized at the end of the chapter(Section 1.4).

1.2 Related Work in Video Coding/Adaptation

The progress of video coding research can be well represented via the development of video coding standards by ITU-T Video Coding Experts Group (VCEG) and ISO/IEC Moving Picture Experts Group (MPEG). Figure 1.2 describes the road map of the major coding standards, where H.261 [82] is recognized as the first practical video coding system, and H.264 [84] is the latest advanced coding standard. Besides these coding standards there are also several significant progresses reported in the research literature. In this section we provide an overview of these emerging coding standards and research trends.

1.2.1 H.264/Advanced Video Coding

H.264 is the latest ITU-T/MPEG joint video coding standard [84, 100], also known as MPEG-4 part 10, or Advanced Video Coding (AVC)). The main goal of H.264 is to enhance compression performance and provide a network-friendly video representation considering conversational (video telephony) and non-conversational (storage, broadcast, or streaming) applications. A family of advanced techniques (such

Year	ITU-T	MPEG	Application	Format	Bit Rate	Techniques
2003	AVC/H.264/MPEG-4 Part 10					
1998		MPEG-4	Distribution over network	CIF QCIF	8kbps ~ 1.5Mbps	Error resilience, synthetic scene, object, ¼ pel MC
1995	H.263 (+, ++*)		Video conf. over Internet / Wireless	SQCIF~16 CIF	8kbps ~ 1.5Mbps	½ pel MC, Variable Block Size, 3DVLC, PB Frames
1994	H.262	MPEG-2	Distribution on DVD / DTV	HDTV	3Mbps ~ 10Mbps	5.1 Audio, AAC, interlace, scalable
1992		MPEG-1	Distribution on VCD/WWW	CIF	1.5Mbps	MP3,GOP, random access, PB frames,
1990	H.261		Video conf. over ISDN	CIF QCIF	Px64kbps	Intra-, inter-MB pel-Level MC, VLC

*: H.263+ and H.263++ were finalized in 1998 and 2000 respectively.

Figure 1.2: Road map of video coding standards [93].

as multiple frame referencing, variable block size motion estimation, integer transform, context adaptive binary arithmetic coding, etc.) have been integrated into H.264. Figure 1.3 provides some experiment results comparing the R-D performance for several coding standards at different bit rates. It is evident that H.264 achieves significant improvement in R-D efficiency compared to existing standards ($2 \sim 3$ times better than MPEG-2 [71]).

The coding structure of H.264 includes the video coding layer (VCL), which carries encoded video signal, and the network abstraction layer (NAL), which encapsulates the VCL and corresponding header information appropriate for a diversity of transport layers or storage media [100]. Though H.264 has been standardized since 2003, many important research topics related to practical issues emerge, such as improving encoding quality performance [86], reducing the computational complexity [11, 13, 35], and facilitating the hardware implementation [111]. We will provide further discussion about these topics in Chapter 5. H264Perf.eps

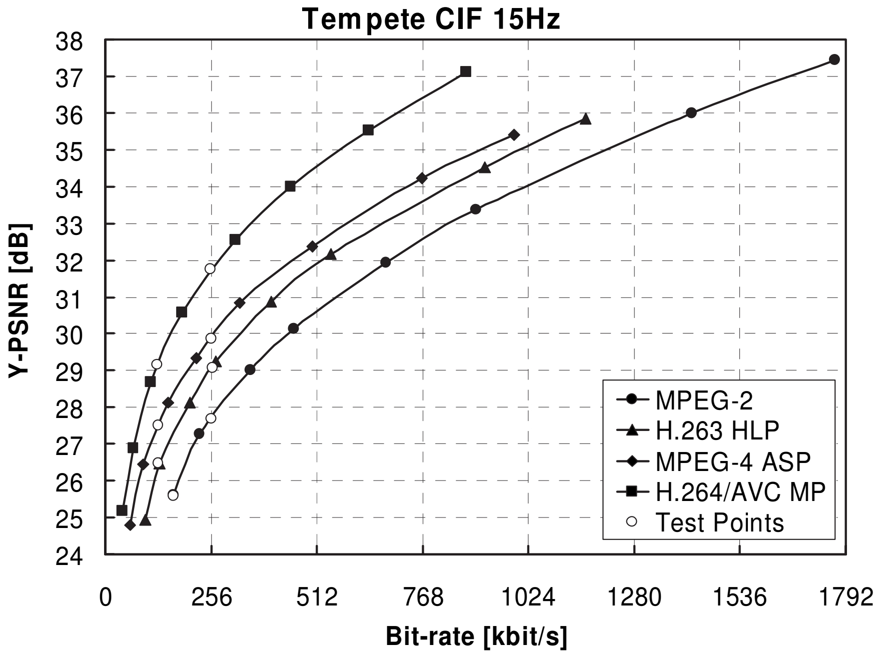


Figure 1.3: H.264 outperforms its predecessors by 2 ∼ 3 times [100].

1.2.2 Video Adaptation

Video adaptation is the process that reshapes an encoded video stream into a new version in order to match new resource constraints (e.g., bandwidth and resolution) or user preferences [1]. Though video adaptation research has appeared in early efforts of compressed-domain bit rate reshaping and video composition [8, 51], it remains as an active topic, especially in response to the emerging applications of UMA. Depending on the location where the operation is carried out, different types of adaptation techniques have been proposed in the literature. For example, requantization of transform coefficients [99], frame dropping (FD) [66], DCT (discrete cosine transform) coefficients dropping (CD) [22] and resolution reduction [106] are

[1]For non-scalable video codecs, this procedure is well known as video transcoding [4]. Scalable video coding usually does not need an explicit transcoding procedure because the scalability is naturally embedded. This thesis uses the term *adaptation* in order to address both cases.

commonly used. More detailed discussion involving adaptation for UMA can be found in [4]. When combining more than one type of adaptation operations, we can form multi-dimensional adaptation (MDA). MDA provides greater flexibility to satisfy a wider range of resource constraints, but the issue of optimal operation selection also arises. More discussion about MDA will be included in Chapter 2.

The adaptation methods mentioned above are mainly designed for block based non-scalable MC-DCT (hybrid) video codecs. A codec is considered as scalable if it generates flexible bit-streams that can be partially decoded in order to satisfy different bandwidth conditions. As a comparison non-scalable video codecs generate bit streams that are optimized for one specific bandwidth, lacking the scalability for other resource conditions without explicit transcoding. Therefore scalable video codecs usually provide greater convenience in reshaping the bitstream with better maintained video quality compared to non-scalable video adaptation operations (such as requantization). A well-known example of scalable adaptation solution is MPEG-4 Fine Granular Scalability (FGS) [64]. Recent research activities [9, 15, 33, 63, 67] in wavelet/subband video coding greatly expand the usability and performance of scalable video coding. As part of the continued MPEG standard activities, a subgroup in MPEH-21 is currently developing a new scalable video coding standard [83].

1.2.3 Motion Compensated 3D Subband/Wavelet Coding

Motion cmpensated 3D subband/savelet coding (MC-3DSBC) technology undergoes rapid development recently. The success of SBC has been shown in static image coding such as JPEG2000 [17]. The extension of SBC to motion pictures relies on the efficient processing of motion compensation (MC) along the temporal dimension, called motion compensated temporal filtering (MCTF). Approaches based on MC-

3DSBC have been proposed to address this issue. One early important progress in 3D subband coding of video is the work by Ohm [53], where a non-recursive MC-3DSBC scheme is proposed. Choi and Woods extended Ohm's work by proposing a 3D subband-finite state scalar quantization (3DSB-FSSQ) coding system for video [15]. Both techniques employ unidirectional motion compensation. Chen and Woods [9] further advanced the work by using bidirectional MCTF. The promising performance of the subband coding system also comes from the use of embedded coding architecture. Since the nature of subband filtering provides a layered hierarchical representation, it can efficiently organize the transform coefficients based on their energy distribution and correlations in different layers. The embedded coding explores two basic properties of subband coding coefficients: first, coefficients in the same spatial position from different subbands share similar energy behaviors; second, the coefficient magnitude of a higher level subband is statistically smaller than the magnitude of the ones at the same spatial location in a lower level subband. From embedded zerotree wavelet(EZW) [67] to motion compensated embedded zeroblock coding (MC-EZBC) [9], embedded coding systems have made significant progress [9, 33, 63, 67]. More discussion about this topic is presented in Chapter 4.

With all of these efforts, the latest MC-3DSBC codecs have been shown to achieve excellent coding performance comparable with the non-scalable video codecs such as H.264 [9]. Besides, MC-3DSBC has remarkable performance in video adaptation. Though traditional non-scalable video codecs support conceptually similar adaptation operations (such as frame dropping and resolution downsampling), they usually need considerable extra computing cost, and suffer from the well-known problem of error propagation that degrades the video quality [22, 105]. As a comparison, MC-3DSBC can provide more convenient implementations while yielding better video quality [9]. Because of these advantages, MC-3DSBC based proposal is one major candidate in recent scalable video coding standard effort [77] [2].

1.2.4 Power/Complexity Aware Video Coding

In the rich media era many wireless/mobile devices are capable of generating/consuming video data, whereas the power consumption/computational complexity becomes a serious bottleneck. To address this problem, besides some generic solutions (such as hardware level power saving techniques [23, 75] and wireless transmission power optimization [19, 101]), power/complexity aware video coding provides an alternative promising direction. On the other hand, the latest video codecs such as H.264 and MC-EZBC includes many complex operations to improve the coding performance. Therefore, developing a low-complexity codec becomes a very important issue. Different from other video coding work where bandwidth is the major constraint, power aware video coding incorporates the power/complexity cost in optimizing coding performance. It usually constructs a rate-distortion-complexity optimization framework. Depending on the application scenarios and control parameters, the problem formulation vary [21, 34, 46, 109]. More discussion will be included in Chapter 5.

1.2.5 Other Topics

There are several other significant research activities in the literature. 3D-video coding [76] is targeting at providing video bit streams with more one viewpoint. This is a useful technique for some emerging applications such as interactive video based virtual reality touring and stereo video. Distributed video coding [25, 56] novelly extended conventional Wyner-Ziv coding (that is used in channel coding) and proposed a video coding architecture with a lightweight encoder and a heavy-

[2] The latest standard activity adopted the extension of H.264, instead of wavelet-based solutions, as the reference of scalable video coding [83]. Nevertheless, some critical techniques used in MC-3DSBC, such as MCTF, is also adopted by this extension.

weight decoder that accomplishes most of the complex computing (such as motion estimation). Although these topics are not directly related to our thesis work, the utility-based adaptation framework and the content-based operation prediction approach are general, offering potential of extension to the new generations of video representation, such as 3D coding.

1.3 Open Issues and Motivation

1.3.1 Problem Formulation

Most of video coding systems include some R-D optimization components solving problems formulated as below.

$$\sum_{\mathbf{P}} D(\mathbf{P}) \qquad (1.1)$$
$$s.t., R(\mathbf{P}) \leqslant R_T$$

where $\mathbf{P}$ represents the control parameters (CP) that eventually determine the final video quality and bit rate. Typical CPs include quantization parameter (QP), motion vector, block mode, etc. D is the distortion introduced by the encoding process. R is the bit rate of the encoded video and is the target bit rate. The solution of the above problem aims at finding the optimal CPs for each coding unit in order to minimize the average distortion while satisfying the bit rate constraint. If we incorporate additional resource constraints, quality evaluation and control parameters, Equation 1.1 can be extended and a generic resource constrained optimization problem can be formulated as:

$$\sum_{\mathbf{P}} U(\mathbf{P}) \qquad (1.2)$$

$$s.t., \mathbf{R}(\mathbf{P}) \leqslant \mathbf{R}_T$$

where utility U is defined as any applicable quality evaluation (such as peak signal-to-noise ratio or PSNR, subjective evaluation, or even delivery of semantic meaning [73]). Resource $\mathbf{R}$ is defined as any available supply or support such as bandwidth, power consumption and computing complexity. Equation(1.2) is applicable to both video coding and adaptation scenarios. Equation(1.1) can be considered as a special case of Equation(1.2). This extension, though straightforward, makes it more difficult to find the optimal solutions. Conventional R-D optimization solutions adopts computable distortion measures (such as SNR or mean squared error, MSE), from which analytical solutions or efficient iterative search of solutions are available. When the utility and resource are extended to include diverse types (such as subjective perceived quality or coding preference), conventional analytical solutions no longer work. For example, as we shall see in Chapter 2, if other quality evaluation metrics rather than PSNR and control parameters rather than QP are adopted, analytical modelling of relationships between resources and utility becomes difficult, if not impossible any more. Therefore, alternative approaches are called for.

1.3.2 Open issues that motivate this thesis work

Specifically, we are motivated to attack the following problems:

1. *Multiple dimensional adaptation (MDA).* MDA combines more than one type of independent adaptation operations, offering greater flexibility in adapting the video in meeting the specific resource constraints, such as bandwidth, power, and display size [3]. However, it also creates many new challenging prob-

[3]Conventional R-D framework also utilizes several control parameters during video coding. E.g.,

lems. First, measurement of video utility in the multiple-dimensional space is a much harder issue. For example, how do we evaluate the subjective quality of videos of different frame rates, spatial quality, and/or spatial resolutions. Furthermore, finding the optimal adaptation operation in the multi-dimensional space is a much harder problem than that in single dimensional space.

2. *Power/Complexity aware H.264 coding.* Emerging video coding standard H.264 achieves significant advances in improving video quality and reducing bandwidth, but at the cost of greatly increased computational complexity at both the encoder and the decoder [38]. Though there are quite a few work providing low complexity solutions from hardware [10, 111] and encoder aspects [13, 35, 69, 85, 103], how to reduce the power consumption/computational complexity from the decoder side is less addressed. We are interested in this issue because we believe in the near future wireless/mobile devices will be typically used for video consumption, rather than production. How to reduce the computational complexity in decoding complex compressed streams such as H.264 is of critical importance. For non-interactive application scenarios such as on-demand access, improving the quality and efficiency of the client-side decoders is of great interest.

1.4 Contributions and Overview of the Thesis

In this section we summarize our contributions to solving the open problems discussed in Section 1.3 and outline the thesis organization.

besides QP, there are also motion vectors, block mode and Lagrangian parameter. Nevertheless, in practical implementation these parameters are all tied to QP, and only MSE or SNR are used for quality metrics. Therefore, in this thesis such case is not considered as multiple dimensional approach.

1.4.1 Thesis Contribution

1. *MDA behavior investigation.* We carefully analyze the MDA properties for both non-scalable video codec and scalable video codec. In particular, we combine frame dropping (FD) and coefficient dropping (CD) in MPEG-4 simple profile and define MPEG-4 FD-CD adaptation; we also use one of the latest MC-3DSBC codecs, MC-EZBC, and construct MDA capable of joint SNR-temporal scalability. The properties of each adaptation dimension is analyzed. Some specific issues such as the rate control in MPEG-4 FD-CD and the SNR optimization in MC-EZBC are addressed. These provide the basic understanding and necessary platforms for exploration of other topics described below.

2. *Subjective preference exploration in MDA*. Instead of traditional MSE based objective quality evaluation, for MC-EZBC we launch thorough subjective quality evaluation in order to understand the relationship among video content, MDA operation and perceptual quality. Specifically, we conduct experiments to study human preference between temporal rate and spatial details for different types of video content over different ranges of bit rates.We apply formal statistical analysis tool to process the experiment data and discover important rules about how the subjective perceptual quality is influenced by the video content, available bandwidth and MDA operations. To our knowledge this is one of the first work that studies the subjective quality characteristic under multiple dimensions of adaptation.

3. *Utility function (UF) based video adaptation.* We propose a UF based video adaptation framework. UF is derived from the adaptation-resource-utility (ARU) space mapping and is demonstrated as an efficient tool in selecting

MDA operations, especially in a real time manner. We specify the way how UF is defined, generated and represented, and how it can be used in MDA. Our UF-based description for video adaptation has been submitted to MPEG-21 as a contribution, which has been accepted as part of the final standard for description scheme for MPEG-21 Digital Item Adaptation (DIA).

4. *Content-based optimal adaptation operation selection.* We propose a general classification-based prediction framework for selecting the preferred MDA operations. Content features are computed from the compressed video streams and are used as input to a classifier which maps the input video to one of the trained classes, from which UF's and optimal adaptation operations of the new video are automatically predicted. Such a system uses machine learning approaches to discover the common characteristics of videos of the same content type, thus avoiding the need of constructing the analytical models of the utility function, which is difficult in the MDA case. A machine learning based method is applied where the low level content features extracted from the compressed video streams are employed to train a framework for the problem of optimal MDA operation selection, thus avoiding the intractability in building up analytical modelling. The proposed framework is extensively validated using MPEG-4 FD-CD and MC-EZBC platforms with excellent performance, base on both classification accuracy and prediction precision.

5. *Complexity adaptive motion estimation and mode decision for H.264 coding.* Motion estimation (MC) is the single component that dominates the complexity of H.264 decoding. Instead of modifying the MC component in the decoder directly, we have developed a systematic solution by constructing a rate-distortion-complexity (R-D-C) optimization framework in the encoder

side. Our approach is novel - encoded videos from our encoder much lighter weights (in terms of decoding complexity) at almost the same quality level. In addition, we have developed a novel complexity control method to meet the target complexity level and maintain consistent complexity throughout the entire video stream. Our algorithm monitors the complexity consumption status and effectively predicts the appropriate control parameter to be used in the R-D-C optimization procedure. Our simulation results show excellent results in reducing complexity while keeping the video quality more or less intact.

1.4.2 Thesis overview

The remaining of this thesis is organized as follows. Chapter 2 provides a general description of MDA. Thereafter the UF is defined and the usage of UF is explained. Based on these, the content based prediction framework is introduced. Chapter 3 and Chapter 4 apply the UF based adaptation framework to video encoded in MPEG-4 and MC-EZBC respectively. The performance of content based MDA operation prediction is presented. Chapter 5 includes our complexity adaptive H.264 decoding solution, including the derivation of the R-D-C optimization, complexity modelling, complexity control, and the experiment results. Chapter 6 includes conclusions of the thesis work and discussion about future research directions.

Chapter 2

Utility Function Based Video Adaptation and Content Based Prediction

As introduced in Section 1.2.2, there exist many adaptation techniques in the literature. Nevertheless, joint exploration of multi-dimensional adaptation is much less addressed. One reason is the lacking of an efficient and systematic framework in selecting optimal MDA operations. In this chapter we provide a general description of our proposed solutions for this open issue, namely utility function based video adaptation and content based prediction. The utility-based framework offers a systematic approach to solving the problem, while the content-based approach makes real-time operation prediction possible.

The remaining of this chapter is organized as follows. Section 2.1 reviews the existing video adaptation operations and discusses the characteristics of multiple dimensional adaptation. Section 2.2 and Section 2.3 introduce the utility function based video adaptation and content based prediction framework respectively. And Section 2.4 includes brief summaries.

2.1 Multiple Dimensional Adaptation

2.1.1 Common video adaptation operations

In UMA applications, video adaptation is an essential operation so that one single version of video stream can be altered to match various resource constraints and user preferences. Figure 2.1 illustrates some common adaptation approaches and their locations in the decoding pipeline. These approaches come with different effects on quality and computational complexity. Specifically, the later the adaptation is executed in the decoding process, the better the video quality is, but on the other hand the more complex the operation is. The most straightforward "decoding and re-encoding" operation decodes the compressed video back to the uncompressed domain and then tries to optimize the quality of the new encoded stream. The optimized quality comes with the high cost of computational complexity. To reduce the computation cost, some work proposing reuse of the motion vector information is discussed in [72]. However, the computational workload for other components such as inverse DCT (IDCT) and rate-distortion optimization is still non-ignorable. On the other hand, frame dropping has the least complexity, but the temporal resolution of the video stream is sacrificed. Such tradeoff between computational complexity and video quality needs to be carefully assessed when considering a practical application. In this section we provides a quick review on the common used adaptation operations.

2.1.1.1 Requantization

Quantization is widely applied in lossy compression to reduce the entropy of the source signal, thus further facilitating other source coding techniques such as entropy coding. In video coding it is the procedure mapping continuous input DCT

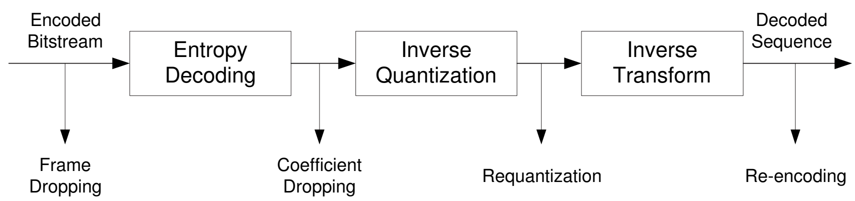


Figure 2.1: Common adaptation approaches and their locations in the decoding procedure.

coefficients into discrete quantized levels. Quantization can be represented as:

$$y = Q(x) \tag{2.1}$$

where y is the quantization level, x is the source signal (DCT coefficients), and Q is quantization mapping function. Depending on the optimization criterion and signal statistical characteristic, the mapping function can be uniform or non-uniform, with or without dead zone [26, 55]. Requantization is one of the most common adaptation operations [68, 89, 99]. It is the procedure to use a new (usually coarser) quantization mapping function to quantize the reconstructed coefficients. I.e.,

$$y' = Q_r\Big(Q^{-1}(y)\Big) = Q_r\Big(Q^{-1}\Big(Q(x)\Big)\Big) \tag{2.2}$$

where Q_r is the new quantization mapping function. It is a design issue to choose appropriate requantization mapping function Q_r and reconstruction function Q^{-1} to optimize the process subject to some cost function [55]. Typical cost functions are MSE and maximum a posteriori (MAP) [68]. To guide the design choice of Qr and Q-1, sometimes source signal models are used, source signal model can be employed to leverage this problem. e.g., high frequency (AC) DCT coefficients are often modelled as two-sided Laplacian distribution [14].

Requantization provides opportunities for achieving good video quality because of the freedom in optimizing the requantization process. Nevertheless, its drawback is the high computational complexity in obtaining the optimal solution, which is not appropriate for real time adaptation application. Also the source signal modelling mentioned above does not necessarily match the actual signal statistic and thus may result in suboptimal solutions.

2.1.1.2 Coefficient dropping

The concept of dynamic rate shaping (DRS) was proposed in [22] to address the issue of bit rate reshaping. Coefficient dropping (CD), the procedure to selective keep only part of the DCT coefficients, is one of the key components in DRS. The philosophy behind CD is that the low frequent coefficients contains more signal energy and thus aremore important for perceptual quality. The higher frequent coefficients, on the other hand, are either less important to perceptual quality or contain much less energy. Therefore, we can remove some high frequency coefficients without significantly degrading the video quality. Figure 2.2 [22] shows two basic ways to achieve CD within each DCT block: constrained and unconstrained CD, where the former throws away the coefficients beyond a single truncation point, and the latter has the freedom to cut any set of coefficients based on some optimization criteria. The unconstrained CD has been found to require much higher computational cost, however does not necessarily achieve a higher video quality [22].

Compared with requantization, CD sacrifices the video quality because the quantization step is untouched and thus it has less freedom in optimizing the performance. As a tradeoff CD requires much less computing cost, making it a more suitable method for real time adaptation.

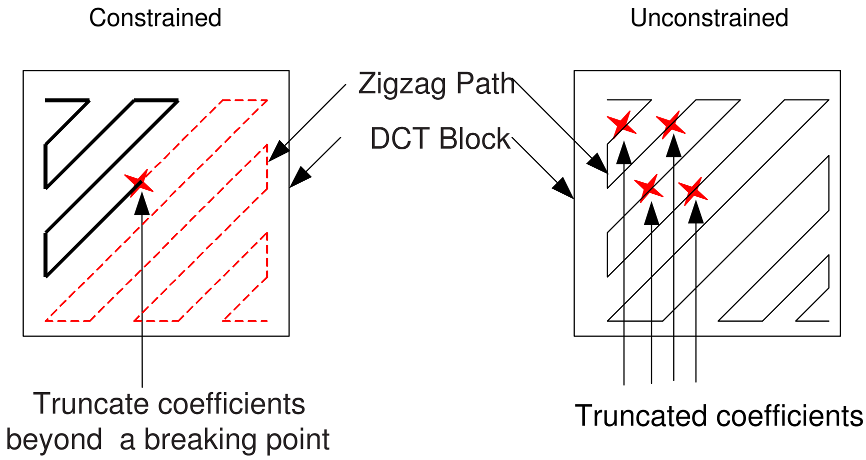


Figure 2.2: Constrained and unconstrained CD.

2.1.1.3 Frame dropping

Frame dropping (FD) is the procedure to remove certain frame(s) from a video sequence in order to achieve bit rate reduction. FD is equivalent to temporal resolution downsampling. As a dramatic way to reshape target bit rate, FD is very useful in the scenarios where the end devices have limited computing capability in supporting higher frame rate. In order to preserve the video quality, a content-based approach has been proposed to be combined with FD so that only the less important frames are dropped [43]. In addition, for inter coded frames, the motion vectors pointing to the skipped frames are no longer valid. To avoid this one way is to apply FD only to the frames that are not referred by other frames [36], e.g. B frames. Otherwise the motion vectors need to be re-estimated [20]. Similarly the residual errors might need re-estimation [24, 107]. More detailed discussions of FD can be found in [89].

2.1.1.4 Resolution downsampling

Resolution downsampling are useful for applications where the devices, especially wireless/mobile ones, have limited display resolution. Resolution downsampling can be implemented in either pixel domain or compressed domain and the computational complexity varies. The former is a straightforward method which involves an re-encoding procedure. The latter tries to accomplishes down-scaling directly in the DCT-domain without converting DCT coefficients back to pixel values so that real-time implementation is feasible. [106]. Similar to frame dropping, the motion vectors after the downsampling may need re-estimation (refinement) and the DCT coefficients may need re-organization to match the new resolution [41, 106].

2.1.1.5 Error propagation issue

All of the adaptation operations discussed above suffer from the well known phenomenon of error propagation, meaning the distortion will be accumulated and the quality will be degraded when adaptation operation is applied over multiple frames. The is because the adaptation introduces changes to the encoded data, which may have been used as reference, such as the reference in the motion estimation process. The modification, if not adjusted, will cost distortion which will be propagated over frames. If not addressed correspondingly, the distortion will be accumulated and exacerbated along with the operation. Therefore, an error compensation process is necessary in order to improve video quality [22, 68, 105].

2.1.1.6 Scalable video adaptation methods

Scalable video codecs implicitly support adaptation operations - a scalable video stream can be truncated at any arbitrary point or certain pre-defined points in order to adjust the bitrate, frame rate, or spatial quality of the video stream. Conceptually, it achieves the same effects as various adaptation methods mentioned above. However, it does not cause error propagation because of the special coding structure used in the scalable encoder. On the other hand, scalable codecs have not been widely used in practice so far since they require adoption at both the encoder and decoder side. Conventional non-scalable decoders cannot accept bitstreams generated by the scalable encoder. A typical example of scalable adaptation solution is MPEG-4 Fine Granular Scalability(FGS) [57], which applies bitplane based coefficient coding so that the fine granularity can be achieved by bitplane truncation. Recently the efforts in wavelet/subband video coding greatly expand the usability and performance of scalable video coding. More details about scalable video adaptation will be discussed in Chapter 4.

2.1.2 Multiple dimensional adaptation

In this thesis, we are interested in adaptation systems that use combinations of more than one type of adaptation operation, resulting in Multi-dimensional Adaptation (MDA). MDA offers greater flexibility in optimizing the quality-complexity tradeoff while satisfying the resource constraints. We are particularly interested in the MDA which combines independent adaptation operations. As an example the motion re-estimation associated with FD or resolution downsampling will not be considered as a dimension of operation because it usually facilitate improving the SNR quality. Moreover, adaptation operations to be combined should be of different aspects. For example, requantization and CD both yield coarser quality in each frame. Combination of these two operations will not be an interesting MDA. On the contrary, combination of FD and resolution downsampling allows modification of the frame rate and frame resolution at the same time, and therefore is a interesting and promising MDA, especially for mobile devices. In general, the adaptation operations may be categorized into the following aspects:

1. *SNR adaptation*: the adaptation, such as requantization and CD, results in bitstreams with a reduced quality, while with the spatio-temporal resolution unchanged.

2. *Spatial adaptation*: the adaptation results in reduced spatial resolution, namely a smaller frame dimension.

3. *Temporal adaptation*: the adaptation results in reduced temporal resolution, namely a lower frame rate.

A full SNR-spatio-temporal MDA is certainly attractive for its high flexibility. Nevertheless, in practice it involves several difficulties. First, it is difficult to define an adequate quality measurement metrics over so many different dimensions. Videos resulting from different SNR-spatio-temporal adaptations may have different frame rates as well as spatial resolutions. There are no known quality metrics valid for comparing their qualities. Second, the full level of adaptation flexibility also imposes strict requirements on the codec implementation. It is difficult for existent codecs, either scalable or non-scalable ones, to achieve efficient implementations while supporting all three dimensions of adaptation. For example, our experiment [96] using MC-EZBC indicated that SNR-temporal adaptation always outperformed SNR-spatio-temporal adaptation in a wide range of bandwidth no matter using objective or subjective quality evaluation (the videos with smaller resolution was upsampled before the evaluation).

Based on the facts above, in this thesis we will focus on the SNR-temporal MDA operation. Note in some specific scenarios such as delivering videos to the devices with limited display resolution, spatial adaptation should be considered separately. This is beyond the scope of this thesis and will not be discussed. In Chapter 3 and Chapter 4 we will implement SNR-temporal MDA based on MPEG-4 and MC-EZBC respectively.

The of SNR-temporal MDA lies in several folders has multiple advantages.

1. It provides broad flexibility to satisfy a wide range of resource constraint. Temporal adaptation is a dramatic way in reshaping the video bit rate, but it also influences the video quality significantly. On the other hand, SNR adaptation adjusts the video quality in a fine granular manner, but the adjustable scope is limited. Combination of both dimensions not only expands the scope of bandwidth adjustment but also offers effective methods for retaining video quality.

2. It allows choosing from different SNR-temporal combinations based on perceptual quality or user preferences. Our subjective experiment [92] and previous literature results [58] revealed how human subjective preferences of SNR-temporal tradeoff change under different bit rates for different videos. When the target bandwidth is high, smoother motion (i.e., higher frame rate) is preferred. When the target bandwidth goes below certain point, more SNR quality is desired for preserving spatial details. It is very interesting that we have found the bandwidth threshold for preference changes varies for different video content. More details about subjective quality evaluation will be specified in Chapter 4.

3. For video signal with a lower frame rate, the computational cost is much less than the ones with a higher frame rate. Therefore, SNR-temporal adaptation

is useful for controlling the computational complexity, an attractive feature for wireless/mobile applications .

The above benefits of the SNR-temporal MDA come with a challenging issue – how to automatically determine the optimal combination of SNR quality and the frame rate. To our knowledge most existing adaptation techniques either concentrate on optimization of pre-selected adaptation operations, or select the MDA operation in an ad hoc way. We propose a utility function based video adaptation method as a systematic approach to this problem. Recently Gotz and Patel presented a similar framework for MDA independently in [27]. Whereas they did not address the problem of real time utility information generation. We will first provide a description of the utility function based video adaptation, and then introduce the content-based prediction framework.

2.2 Utility Function Based Video Adaptation

2.2.1 Adaptation-resource-utility space

The UF-based adaptation approach fits very well a three-tier server-proxy-client adaptation architecture, shown in Figure 2.3. The adaptation engine deployed in the proxy adapts incoming videos to satisfy dynamic resource constraints that are not known a priori. The role of the UF is to describe the relationship between required resources and resulting video utilities when the video is subject to various adaptation operations in multiple dimensions. For stored videos, UF can be generated offline at the server and sent to the adaptation engine. The engine will then select the optimal adaptation operation based on the information in the UF. For live videos, UF needs to be obtained on the fly through some estimation and update processes.

We specifically propose a content-based prediction method that estimates the UF according to the content features and statistical classification tools. Such real-time prediction methods can be implemented at either the server or the proxy.

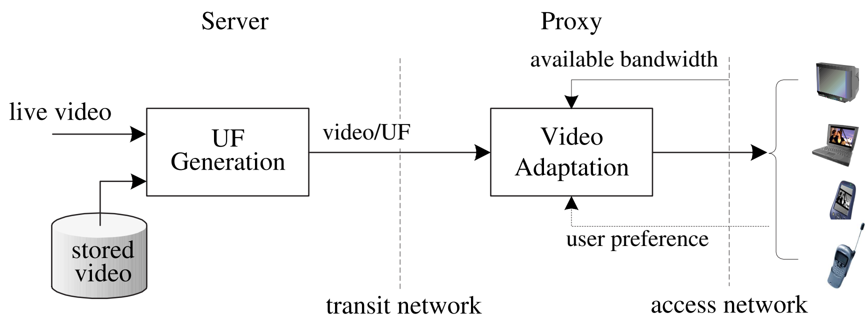


Figure 2.3: A three-tier adaptation architecture using the utility-based framework.

2.2.2 Definition of adaptation, resource, utility and their relations

UF is defined in the Adaptation-Resource-Utility (ARU) space [7], where relationships among diverse types of adaptations, resources (e.g., bandwidth, power, and display) and utilities (e.g., objective or subjective quality) are modeled. We use the term *space* in a loose sense here to indicate the multiple dimensionality involved. Figure 2.4 depicts the notions of ARU involved in a video adaptation problem. The entity, e, refers to the basic unit of video data that undergoes the adaptation process. Adaptation operators are the methods to reshape the video entities, such as requantization and frame dropping. All permissible adaptations for a given video entity constitute the adaptation space. Resources are constraints from terminals or networks, including bandwidth, display resolution, power, etc. Utility represents the quality of an entity when it is rendered on an end device after adaptation, such as PSNR, perceptual quality, or even high-level user satisfaction. The mapping relationship among ARU spaces is illustrated in Figure 2.5. Each operation in the adaptation space will generate a video bitstream with specific utility and resource values. Typically, there exist multiple adaptation solutions that satisfy the same resource constraints, while yielding different utilities. In Figure 2.5, the points in the oval shaped region in the adaptation space indicate such a constant-resource region. Likewise, different points in the adaptation space (the shaded rectangle) may lead to the same utility value. It is such a multi-option situation that makes the MDA problem interesting - we want to choose the optimal one with the highest utility or minimal resource.

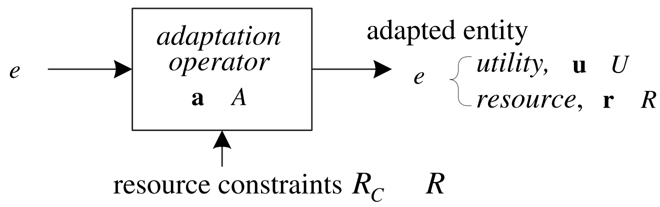


Figure 2.4: Definition of adaptation, resource, and utility spaces involved in video adaptation problems in the utility-based framework.

2.2.3 Utility function

We are interested in describing the relationship between resource and utility associated with each adaptation operation. Such relationship is represented via UF. The right figure in Figure 2.5 shows an example of UF, in which only one dimension is shown in both resource and utility. This is equivalent to the known R-D curve when R is bitrate and D is related to video quality. Each point in UF is associated with one specific adaptation operator, which may include combinations of multiple operations (such as frame dropping and coefficient dropping). Note though UF contains discrete ARU data, it is possible to interpolate along the dimension where fine granular control is possible, such as Requantization and CD. This procedure is illustrated as doted line in Figure 2.5.

It is easy to notice the similarity between R-D and UF framework. Indeed, R-D framework can be considered as a special case of UF if we choose rate for resource, MSE distortion as the utility, and quantization parameter as one dimension of adaptation operation. UF extends R-D into a more generic ARU space. This extension is not trivial because the solution to select the optimal MDA operations might be quite different. We will provide more details about optimal MDA operation selection in Section 2.3.

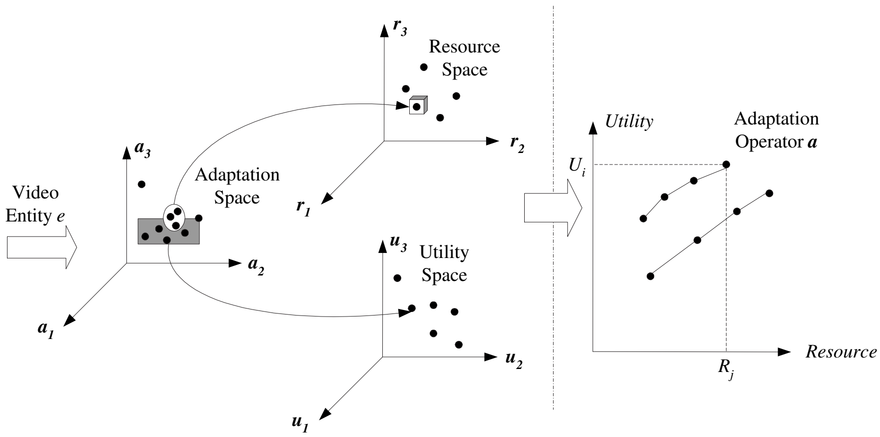


Figure 2.5: Use of UF to describe relations among adaptation, resource and utility.

2.2.4 UF based adaptation

Given UF it is very straightforward to guide choose MDA operation. Consider $\mathbf{a} = (a_1, a_2, ..., a_{|A|})$ is an MDA operation defined in the space $\mathbf{A}$, where each element a_i stands for a constituent adaptation operation and $|A|$ is the dimension of the adaptation space. In general the MDA selection problem can be formulated as:

$$\tilde{\mathbf{a}} = \arg\max_{\mathbf{a}\in\mathbf{A}} U(\mathbf{a}) \tag{2.3}$$

$$s.t., R(\ \mathbf{a}) \leq R_0$$

where $U(\mathbf{a})$ is the utility of the adapted video after operation $\mathbf{a}$ is applied. $R(\mathbf{a})$ is the new resource requirement of the adapted video, and R_0 is the resource constraint implied by the user environment. One example of UF is illustrated in Figure 2.6(a), where the adaptation space consists of two dimensions - a_1 and a_2. a_1 has discrete values and a_2 has continuous values (through interpolation if necessary). SNR and temporal adaptation operations are good examples for a_2 and a_1 respectively. Under the resource constraint R_0, several MDA operations are available: $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$. From UF it is easy to choose the right MDA that maximizes the utility. As a comparison, a simple R-D curve is shown in Figure 2.6(b), which involves only one dimension of adaptation (e.g., quantization).

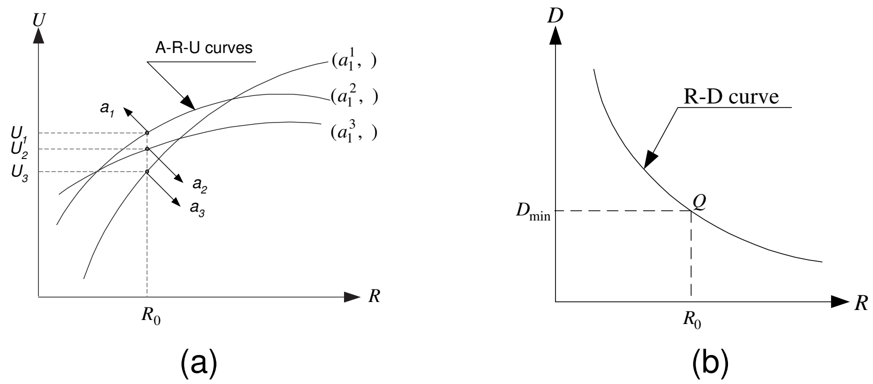


Figure 2.6: (a) an example MDA space showing the relation between utility (U) and resource (R) when the adaptation (A) space involves two dimensions - a_1 has discrete values and a_2 has continuous values. (b) conventional rate-distortion curves consider the signal distortion as utility and often involves only one dimension of adaptation (e.g., quantization).

UF based adaptation is a convenient systematic approach to leverage adaptation decision based on specific ARU space. Actually, UF has been adopted in MPEG-21 standard as part of AdaptationQoS component.

2.2.5 AdaptationQoS in MPEG-21 DIA

MPEG-21 is one of the latest ISO/IEC standard activities. It is an open multimedia framework "defining a normative open framework for multimedia delivery and consumption for use by all the players in the delivery and consumption chain" [80]. In order to provide standardized description for various adaptation related metadata that are employed to support adaptation decision making and specify the resource constraints, MPEG-21 defines digital item adaptation (DIA) in Part 7 [90]. DIA defines a set of tools to facilitate this purpose. Among them AdaptationQoS is the tool to represent the metadata that is used to support adaptation decision-making [52]. UF is one of the components supported in AdaptationQoS. The relationship among MPEG-21, DIA, AdaptationQoS and UF is illustrated in Figure 2.7.

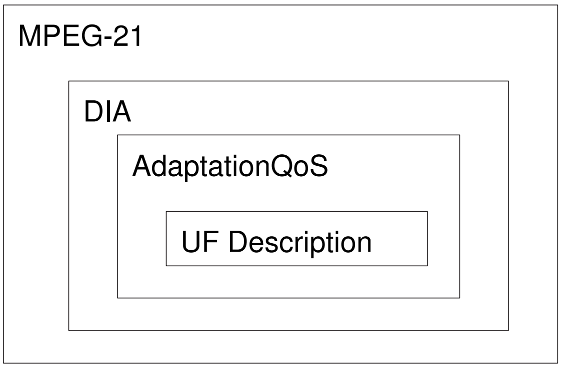


Figure 2.7: The relationship among MPEG-21, DIA, AdaptationQoS and UF Description

Specifically, Figure 2.8 shows the model for an adaptation engine presented in [52]. The adaptation decision taking engine (ADTE) accepts metadata containing decision-taking information (from AdaptationQoS) together with the resource constraint information (from Constraints) and makes decisions about the adaptation operation. The adaptation parameters are transmitted to the bit-stream adaptation engine (BAE) where actual adaptation is executed. UF is one of the methods in AdaptationQoS to provide the description about ARU information. Figure 2.9 depicts an XML schema diagram of the proposed descriptor of UtilityFunctionType under AdaptationQoSType. The exact semantics of the elements in UtilityFunctionType are specified by the elements of IOPin under AdaptationQoSType by referring to the predefined classification schemes [79]. More details about AdaptationQoS in MPEG-21 DIA can be found in [36, 52].

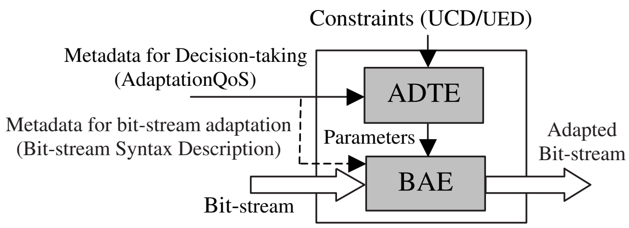


Figure 2.8: Adaptation engine model

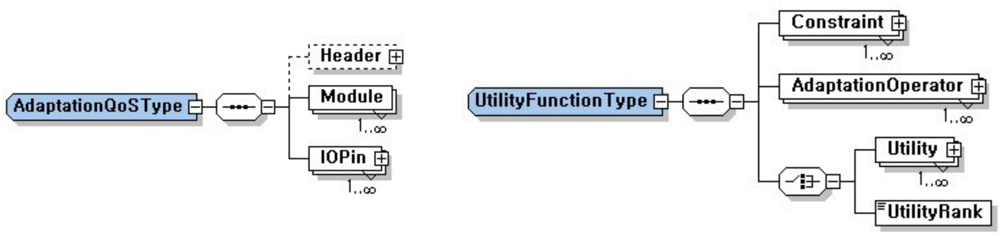


Figure 2.9: Schema diagrams of the descriptors of (a) AdaptationQoSType and (b) UtilityFunctionType

2.2.6 UF generation issue

Though UF provides convenience in selecting optimal MDA operation, in practice the generation of UF is a complex process. It often involves exhaustive computation of a considerable number of adaptation points, each of which requires adapting the video, decoding the obtained bit stream, and evaluating the resource and utility values. This process is very time consuming and typically cannot be done efficiently or in real time. Furthermore, the generated UF is valid only for the specific video clip. Though this UF can be attached with the media data and serve well as the adaptatino metadata (as in the application scenario depicted in Figure 2.8), it can not be directly used by other video clips, especially live sources. Therefore, it is necessary to provide approximate estimation of UF information instead of always physically generating the actually UF. In the next section, we will firstly review the previous approaches for this issue in the literature, then present our content-based prediction framework.

2.3 Content based UF prediction framework

2.3.1 Analytical modelling v.s. statistical estimating

To address the above problem related to the exhaustive computation in UF generation, there are two possible solutions: analytical modelling and statistical estimating. The former has been a popular approach found in the literature. A well-known approach is based on R-D optimization, which has been shown to be an efficient tool for finding optimal coding parameter in the single dimensional case [22, 99]. Some work extended the R-D framework to address the rate control issue in the multi-dimensional space. In [91] an R-D optimization method was proposed by modelling the MSE distortions caused by quantization and frame skipping. In [61], a dynamic programming scheme was used to achieve optimal rate control where frame rate, spatial resolution and quantization step size are jointly considered in modelling the distortion. A distortion measurement is used to estimate the video quality in the full resolution, while some weights are assigned to address the perceptual effects of spatio-temporal scale variation. In [28], variable frame rate coding was realized, where the quantization step size was determined by an analytical distortion model for each frame, and the frame with quantization step size exceeding some threshold was skipped.

Nevertheless, several problems arise when the R-D based methods are applied to solve the MDA case. First, the analytical models often rely on the assumptions that video data follows some statistical distributions such as Gaussian, Laplacian or other variations. The source is usually assumed stationary spatially or temporally. These assumptions are typically invalid in practice. Second, it is not flexible so that the analytical models can be modified in order to take into account different coding structures, utilities (e.g., subjective measures), and resources (e.g., power). Third, it is difficult to come up with an efficient analytical formulation that includes various parameters involved in MDA, and describes the relations between resources and video quality. The formulation can be done in conventional one-dimensional adaptation (e.g., quantization), but becomes difficult for the MDA case. Because of the aforementioned challenges, the power of analytical modelling (especially R-D optimization) approaches is limited in solving the MDA selection problems.

Besides the analytical modelling, another type of methods for selecting the MDA operation is the statistical estimation approach. In [5], a content-based classification system was developed to accelerate the generation of the utility function that characterizes the relation between the approximated subjective quality and the bitrate. In [57], a classification paradigm was proposed to choose from different FGS coding options according to cost functions defined on some objective model of the perceptual quality. In [58], the videos were categorized into different classes using content features extracted from encoded bitstreams and some rules from subjective evaluations were applied to allocate bit rates between temporal layers and spatial layers in MPEG-4 FGS coding. Instead of analytical derivation of the optimal adaptation that achieves the highest utility, these methods are based on the principle that videos can be mapped into distinct categories, each of which comprises videos sharing consistent adaptation behaviors. The adaptation behaviors characterize the utility, required resources, and their relations with various types of adaptations. For example, the conventional R-D curve characterizes the behavior of a video in response to different levels of quantization. Given a new video, compressed-domain features extracted from the bitstream are used to classify the video into a previously learned class, from which the optimal adaptation is predicted. These features include domain-specific knowledge such as minimum achievable bandwidth for particular codecs, and the low level content features describing video characteristic such as motion intensity and texture complexity. Note such a prediction paradigm is fundamentally different from the conventional optimization approach based on analytical derivation, such as Lagrangian optimization. The predicted operation may match or differ from the actual optimal operation. The performance is measured according to the percentage of times when predictions match the ground truths.

The advantages of statistical estimating approach are multifold. First, it is not necessary to rely on an analytical model or empirical relational curves like those used in the R-D optimization framework to relate the impact on subjective quality of the various MDA operations. Alternatively, only the statistical analysis of the correlations between video features and effects of adaptation in terms of resources and quality is necessary. Secondly, the above statistical analysis can be performed either offline or online through classification-based prediction, which can be done efficiently by lightweight feature extraction and classification. Lastly, videos in the same application (e.g. video conference) are likely to share consistent properties and thus make accurate classification and prediction possible.

Figure 2.10 depicts different application scenarios where UF information is generated using different approaches. For an online application (e.g., broadcasting of live events), statistic based prediction offers a light-weight solution suitable for real-time implementation. For offline applications (e.g., delivery of archived videos), both statistical based prediction and non-statistical-based methods are possible, although the former still offer great benefits in terms of implementation efficiency. The generated UF is described using AdaptationQoS syntax so that it can be conducted using DIA tools (see Figure 2.8).

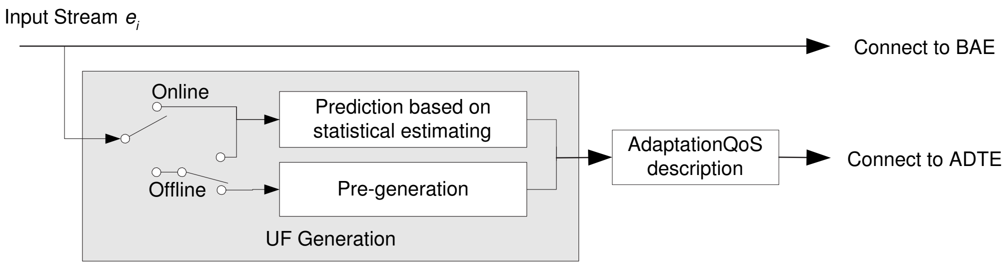


Figure 2.10: Different application scenarios where UF information is computed.

2.3.2 Content-based UF prediction: system architecture

The detailed mechanism of the proposed content-based prediction framework is shown in Figure 2.11. It can be roughly categorized to an offline training route and an online processing route. The former includes the modules for the class definition and classifier learning, and the latter mainly involves classification and prediction. If necessary, regression refinement can be included in both modules. For the class definition part, firstly a pool of video clips is collected as the training data. For each video clip, the compressed-domain features (including domain-specific knowledge, if available, and low level content features) are extracted, so are its ARU information describing the relations between MDA operations and associated utilities/resources. Then, the videos are grouped into distinct categories based on domain-specific rules or domain-independent unsupervised clustering. Videos in the same class are represented by a unique class label and associated with distinctive adaptation behavior metadata. Given the class definition and labelled training data, machine learning techniques are used to learn statistical classifiers and regression model (if any) for mapping video features to corresponding classes and regress values. The learning procedure can be as simple as applying domain knowledge directly, or through standard pattern recognition methods such as support vector machine (SVM). Such classifiers and regression models are then used in the online processing routine to classify the incoming video according to its content features, and predict the corresponding UF for the specific video, which will be sent to the adaptation engine for selecting the MDA operation.

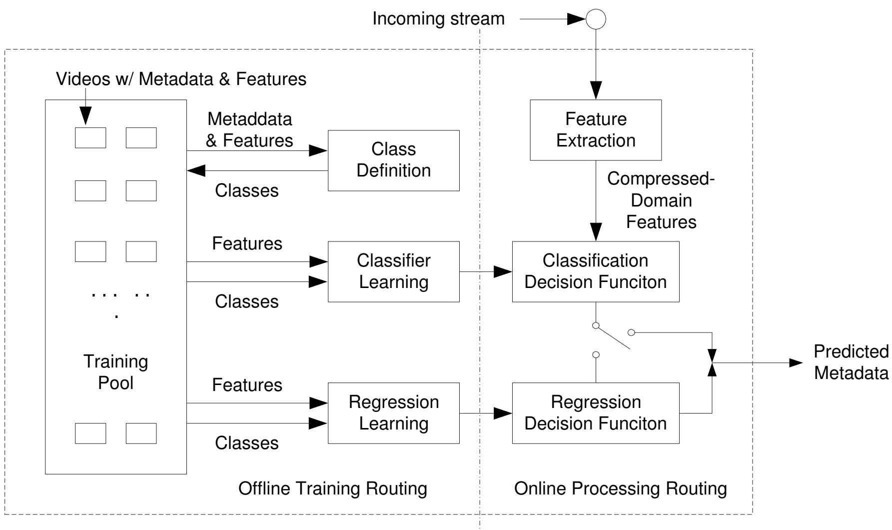


Figure 2.11: Content based UF prediction framework.

2.3.2.1 Content feature extraction

As in any statistical based machine learning, content feature extraction is a standard step. If too many features were included, we would inevitably import noises and result in overfitting. On the other hand, an insufficient feature set would not provide enough information for accurate prediction. It is important to investigate what video features among a vast myriad of possible choices contribute most to the content-based prediction approach. Two facts should be considered:

1. In order to reduce computing cost and support real time scenario, the feature extraction procedure should be implemented in a light-weighted manner. Under this criterion, some pixel domain visual features, such as color and contour, can not be used because fully decoding is required. Whereas compress domain features that can be obtained either directly from the encoded bitstream or by partially decoding are good candidates for our purpose.

2. In the same way that subjective video quality is influenced by the human vision system (HVS), the criterion of content feature selection should be associated with the characteristic of HVS. Though still not fully understood, HVS can be roughly categorized into the spatial mechanism and temporal mechanism[1] [87], where the former accounts for HSV sensitivity to the characteristics of spatial variations, especially different spatial frequency signals, and the latter models the masking and response to temporal phenomena, especially different temporal frequency signals. Since we focus on the SNR-temporal MDA behavior, our feature set consists of attributes related to the spatial and temporal characteristics. The ideal content features are those that contains ample information

[1] In the temporal dimension, the HVS also involves the component of motion tracking. But it is difficult to extract content feature approximating this component.

delivering the SNR-temporal characteristic of the video. Typical examples of such features include motion vectors and texture energy. Depending on the codec implementation, the actual features extracted from the bit streams may vary. We will specify the concrete content features when we apply our proposed framework on specific codecs in Chapter 3 and 4.

3. Even with suitable features extracted, the performance of content-based prediction might still suffer from the noise contained in the features or the correlation implicated among content features. In this situation feature selection is necessary in order to boost the prediction performance. This will also be discussed in the coming chapters.

2.3.2.2 Statistical learning methods

We apply statistical learning methods in several critical components in our framework. In the class definition step, the video clips in the training pool are categorized into distinct classes, each of which is associated with unique UF. Unsupervised clustering is the normal approach for this purpose. Also as we shall see in Chapter 4, sometimes certain domain knowledge such as codec-dependent empirical rules can also be employed to simplify this procedure greatly. In the classification step, the incoming video is labelled into one specific class. This is usually done by supervised learning methods utilizing the extracted content features. In addition, statistical regression is used to refine the accuracy of prediction of the UF for each class. One such example is provided in Chapter 3.

It is worth noting that the proposed framework is general, flexible, and not tied to any specific statistical learning methods. We will provide experiment with state-of-the-art statistical learning methods to validate this point.

2.3.2.3 Computational complexity

From Figure 2.11 it is clear that the online processing route involves only feature extraction, classification and regression, all of which can be implemented efficiently. For compressed input video, some features (like coding parameters and bitrates) are readily available in the headers of the encoded bit streams, while others (like motion intensity and frame complexity) can be efficiently extracted from the compressed domain without full decoding. The offline training process may require some intensive computation, but it is outside the online prediction route and needs to be performed once only. More details will be analyzed when specific video codecs are conducted.

2.4 Summary

In this chapter we provide a general description of our proposed UF based MDA and content-based UF prediction framework. We generalize the relationship among adaptation operations, resource constraints and utility values using the ARU space, within which the UF is defined. UF is flexible and useful for choosing optimal MDA operations from multiple options. However, generating UF information is time consuming and not suitable for real time applications. Previous efforts using analytical modelling are surveyed and the drawback of R-D framework in solving MDA problems are analyzed. Based on this a novel content based UF prediction framework is presented and explained in details.

The effectiveness of the above content-based prediction approach greatly depends on whether consistent, distinctive video classes can be defined, whether representative content features can be efficiently extracted, and whether accurate classifiers and regression models can be realized. In order to evaluate the performance, in the next two chapters, we apply this framework to two specific video codecs: MPEG-4 and MC-EZBC. The first one is a popular non-scalable video codec, and the second one is one of the latest scalable codec. Specifically depending on their characteristics their SNR-temporal MDA behaviors are investigated, UF representations are formulated, and the performance in content-based UF prediction are extensively evaluated.

Chapter 3

SNR-Temporal Adaptation of MPEG-4 Video

In this chapter we apply the proposed UF based adaptation and content based prediction framework to MPEG-4 [37], an ISO/IEO standard finalized in 1998 as the successor of MPEG-1 and MPEG-2. We will first investigate the SNR-temporal MDA behavior of MPEG-4, specifically through the combination of frame dropping (FD) and coefficient dropping (CD). Then the UF using FD-CD is constructed and its usability is introduced. The content based prediction framework introduced in Chapter 2 for MPEG-4 FD-CD is derived, and the performance is evaluated thoroughly.

Our experiments focus on MPEG-4 simple profile, the non-scalable part in the MPEG-4 standard. Though MPEG-4 supports scalable coding through Fine Granularity Scalability (FGS), we will leave the discussion about scalable coding in Chapter 4, where we will use a more advanced scalable coding based on MC-EZBC.

3.1 FD-CD adaptation

Figure 2.1 depicts several commonly used adaptation operations applicable to conventional DCT-blocked hybrid codecs. Each has different adaptation freedoms, effects n video quality, and computational complexity. Among them, FD and CD are both efficient and flexible. Combinations of these two operations offers promising operations for reshaping the video stream bandwidth while preserving reasonable video quality. Therefore, we adopt FD-CD combination as our platform to investigate the SNR-temporal adaptation behavior of MPEG-4.

3.1.1 Frame Dropping

Section 2.1.1.3 described the general features of FD. FD is a temporal down-sampling method by dropping some frames, resulting in large reduction of video bandwidth. To avoid the need of re-estimating the motion vectors, we adopt straightforward operations of FD that drop B and/or P frames that are not used as reference frames by other frames. Figure 3.1 illustrates this procedure, where the GOP size is 15 and the sub-GOP size is 3. Each column indicates a frame, with frame type labeled as I, P, or B. The height of each column gives a qualitative indication of the bit rate of the frame. The shaded frames are dropped. To simplify our simulation and validation process, in our implementation the following rules are applied:

- For P frames, always drop from the end of the GOP so that no additional motion re-estimation is needed.
- For B frames, always drop from the first available frame located in the sub-GOP so that there is no ambiguity.

In Figure 3.1 the first B frame in each sub-GOP is dropped. Because no further motion re-estimation is necessary, the remaining frames is simply copied into the target bit stream and can be decoded correctly.

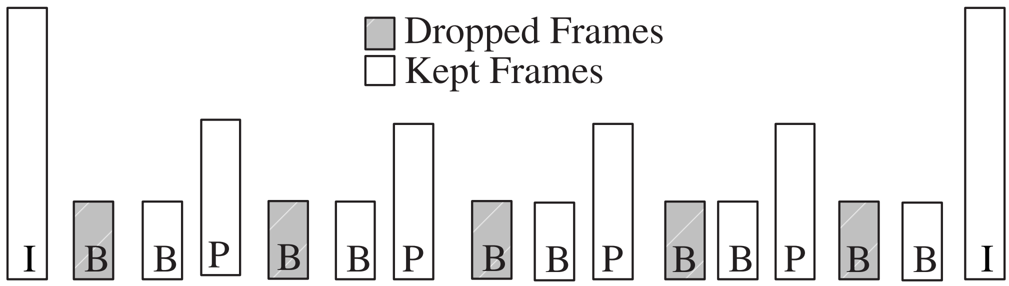


Figure 3.1: Frame dropping based on GOP and sub-GOP structure.

FD provides only coarse approximation to the target rate since the smallest unit of data that can be removed is an entire frame. Adjustment of bit rate at a level finer than the frame cannot be accommodated. As a consequence one issue of FD is the rate control. To overcome this issue, we will complement FD with another operation, CD. We will have an extensive discussion of this issue in Section 3.1.3.

3.1.2 Coefficient Dropping

Our CD adaptation operation is based on the work of DRS [22]. Specifically, we adopt the memoryless constrained model of CD in our system. It has been shown in [20] that such CD model, though not the optimal implementation of CD, is quite close to the optimal solution in terms of video quality. As discussed in Section 2.1.1.2, because the zigzag-scanning pattern of DCT block provides an effective ordering of the DCT coefficients according to their importance, usually the constrained model of CD (see Figure 2.2) results in quality not far from the optimal one. [22] also showed that the memoryless modelling in the R-D optimization, where the accumulated errors caused by motion compensation were ignored and each picture was treated as an independent adaptation unit, did not affect the video optimization performance much (quality difference within 0.3dB).

Figure 3.2 depicts the operation of unconstrained coefficient dropping for one DCT block. The top row is the decoding procedure from the variable length coded (VLC) bits to the pixel domain values (for intra-coded blocks) or MC prediction errors (for inter-coded blocks). CD is applied to the run length symbols after the variable length decoding (VLD). Each run length symbol contains the non-zero coefficient (level), the amount of preceding zeros (run), and an end of block (EOB) flag [78]. For a DCT block with K non-zero symbols, use b^i_{dct} to denote the amount of bits for the i^{th} symbol in the VLC coded symbol. $i = 1, 2, ..., K$. Suppose the k^{th} symbol is located in the truncation boundary, and all of the symbols beyond this point will be dropped. For the k^{th} symbol, it is re-encoded with the EOB flag set to 1. This is illustrated at the bottom row of Figure 3.2. The bit amount b^k_{dct} will be changed to $\tilde{b}^k_{dct}$.

Each frame contains several DCT blocks. Determining the assign truncation point for each of them involves a tradeoff issue between the quality degradation and computational complexity. We consider the following two approaches.

3.1.2.1 Uniform Rate-based CD (URCD)

URCD works as follows: based on the target bit rate, a fixed portion of bits, denoted as η_{bits}, will be truncated from each frame. Practically η_{bits} will be further changed to a practical coefficient-dropping ratio η_{coeff}. Thereafter, the amount of truncated bits is uniformly allocated to each DCT block. Specifically, suppose a video bit stream undergoes reshaping from rate R into R'. $R' < R$. And a given frame has the bit amount B_f for the whole frame and B_c for the AC DCT coefficient bits

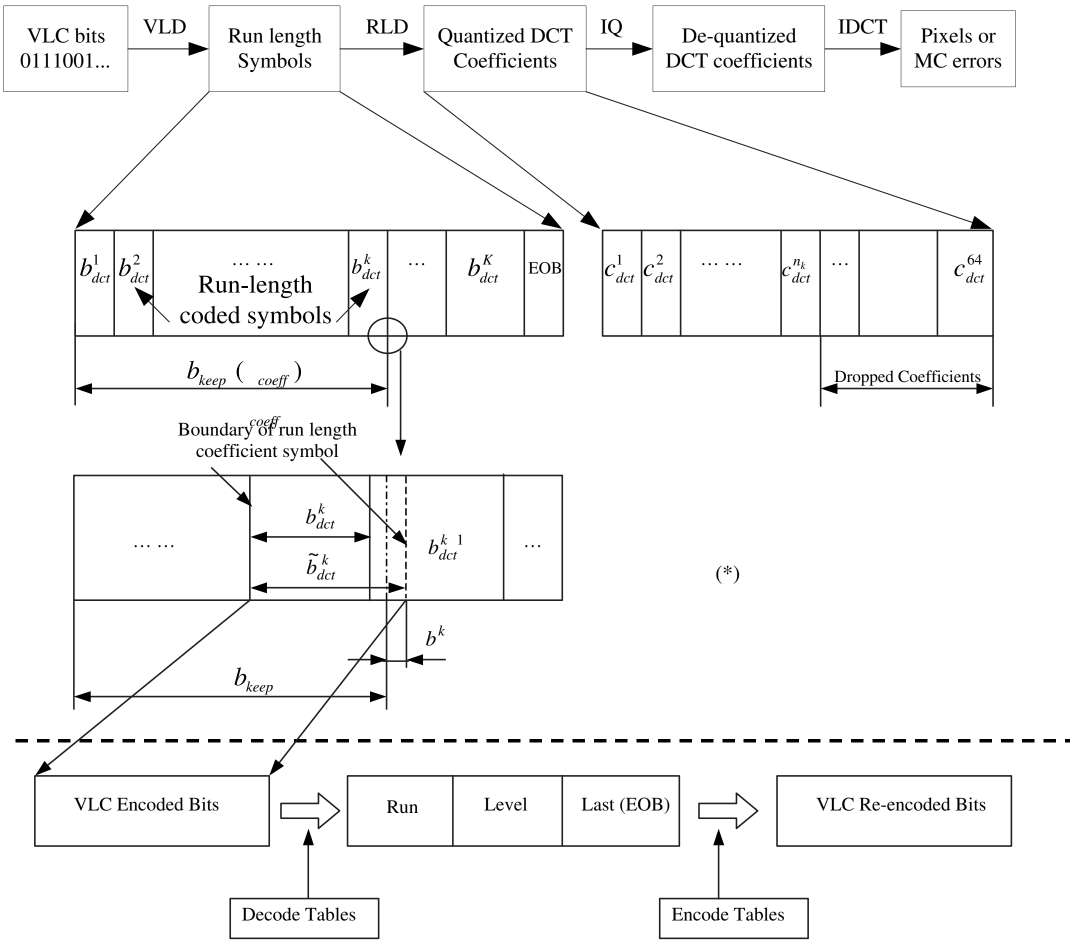


Figure 3.2: Operation diagram of unconstraint coefficient dropping for one DCT block.

only [1]. We get following equations:

$$\eta_{bits} = 1 - \frac{R'}{R} \tag{3.1}$$

$$\eta_{coeff} = \frac{B_f \cdot \eta_{bits}}{B_c} \tag{3.2}$$

The procedure of URCD can be summarized as follows.

[1]only AC DCT coefficients are considered for two reasons: first it is recommended to keep DC coefficient from dropping to maintain baseline quality; second DC coefficients are sometimes coded separately from the VLC coding.

1. Calculate coefficient dropping ratio η_{coeff} from Equation (3.1) and Equation (3.2).

2. Calculate the amount of bits that will be kept b_{keep}:

$$b_{keep} = (1 - \eta_{coeff}) \cdot \sum_{i=1}^{K} b^{i}_{dct} \tag{3.3}$$

3. Find truncation point:

$$k = \arg\max_{k} \left\{ k \middle| \sum_{i=1}^{k} b^{i}_{dct} \leq b_{keep} \right\} \tag{3.4}$$

4. Re-encode the k^{th} coefficient symbol.

5. Calculate the bit amount error:

$$\Delta b_k = b_{keep} - \sum_{i=1}^{k-1} b^{i}_{dct} - \tilde{b}^{k}_{dct} \tag{3.5}$$

The bit amount error Δb_k in step 5 are caused by two facts: coefficient dropping can only be implemented at the boundary of VLC coded symbol, which might not exactly match the required bit amount b_{keep}; re-encoding in step 4 usually gives a different codeword based on the EOB-included 3D VLC [78]. Δb_k will be absorbed by next processed block. This is important to achieve a precise rate control, which will be further discussed in Section 3.1.3.

URCD truncates the original video bit stream with nearly no additional computational cost. The drawback is that as an open loop solution its decoded quality usually cannot be guaranteed because the impact on the overall video quality is not considered.

3.1.2.2 Lagrange Optimization CD (LOCD)

LOCD tries to find the optimal truncation points for each frame. Specifically the following optimization problem is formulated for each frame:

$$\tilde{\mathbf{k}} = \arg\min_{\mathbf{k}} \Big\{ D_{\mathbf{k}} + \lambda R_{\mathbf{k}} \Big\} \tag{3.6}$$

where $\mathbf{k} = (k_1, k_2, ..., k_N)$ is the candidate truncation point set for the frame that has N blocks, $\tilde{\mathbf{k}}$ is the optimal truncation point, λ is the Lagrange multiplier, and $R_{\mathbf{k}}$, $D_{\mathbf{k}}$ are respectively the distortion and rate associated with $\mathbf{k}$. For memoryless constrained CD, $R_{\mathbf{k}}$, $D_{\mathbf{k}}$ are calculated as:

$$R_{\mathbf{k}} = \sum_{j=1}^{N} \sum_{i=1}^{k_j} b_{dct}^{j,i} \tag{3.7}$$

$$D_{\mathbf{k}} = \sum_{j=1}^{N} \sum_{i=k_j+1}^{K_j} \left(x_i^j \right)^2 \tag{3.8}$$

where x_i^j is the non-zero quantized DCT coefficient corresponding to the i^{th} symbol (i.e., the level in 3D VLC) in the j^{th} block. Problem defined in Equation (3.6) can be efficiently solved through one dimension search [59].

In [22], macroblock (MB) based LOCD was adopted. Namely, all of the blocks within a MB will share the same truncation point. Nevertheless in our work the block based search is used. The difference between these two methods can be illustrated in Figure 3.3, where four Y blocks within a MB are shown, and the length of each column stands for the bit amount of each block. It is clear to notice that by using the block-based truncation, a more proportional truncation can be achieved (i.e., the more bits a block has, the more bits will be dropped). So the quality improvement is predictable. Figure 3.3 is the R-D performance for different CD approaches applied on the test sequence *Foreman*. Three results are compared: URCD, block based LOCD and MB based LOCD. As introduced in the figure, MB based LOCD can be considered as a sub-optimum solution of block based LOCD in a smaller searching space. Therefore the quality improvement by using block based LOCD is predictable. For URCD, due to the lacking of distortion optimization, the quality is worse than LOCD. Figure 3.5 provides a zoomed-in comparison of video quality for the first one second of the *Foreman* sequence with the specified format. It is interesting to notice that for some cases URCD even outperformed MB based LOCD, which is a validation of applying block based LOCD.

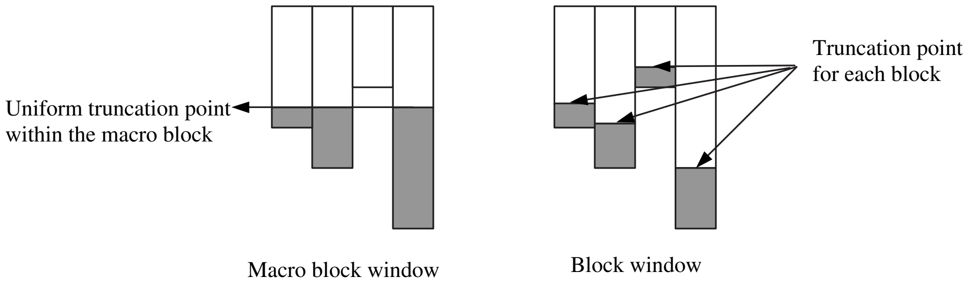


Figure 3.3: MB v.s. block based truncation.

3.1.3 FD-CD combination and rate control

The combination of FD-CD enables SNR-temporal adaptation and extends the dynamic range of the reducible rate while providing fine-granularity video quality control. In FD-CD, some frames are dropped and for remaining frames, CD will be utilized to adjust the target bit rate, as indicated in Figure 3.6.

In FD-CD, the need of rate control comes mainly because of two reasons: imperfect coefficient truncation and dramatic bit dropping due to FD. We will analyze their effects respectively and propose a systematic rate control mechanism.

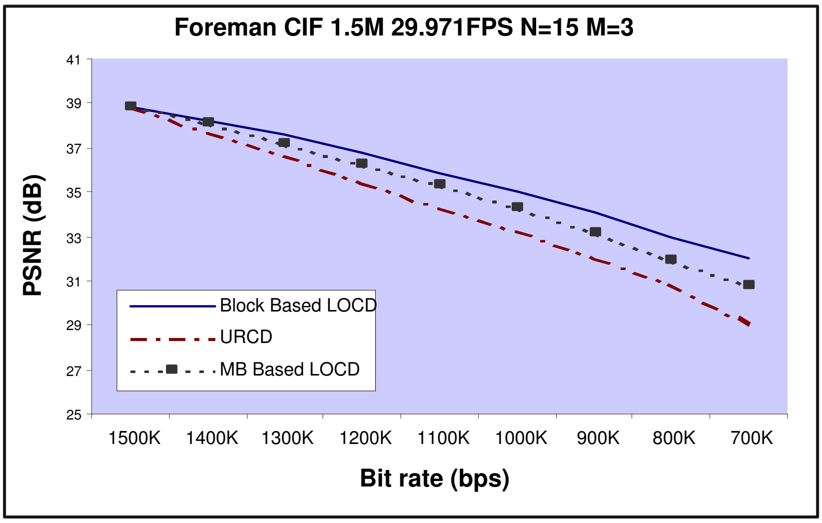


Figure 3.4: R-D performance for different CD approaches

3.1.3.1 Handle imperfect coefficient truncation

Ideally, CD should be able to achieve bit-level precision in adapting a video stream. However, in practice some bit errors occur, called bit drift, mainly due to four reasons.

1. Bit truncation can only be applied atS the run length symbol boundary. For a block with K symbols, the bit drift with respect to truncation point k is:

$$\Delta b_{tr} = b_{keep} - \sum_{i=1}^{k} b_{dct}^{i} \tag{3.9}$$

   where b_{keep} is the amount of coefficient bits after the truncation (see Equation 3.3.

2. The truncation operation is to re-encode the run length symbol that is located at the truncation boundary using 3D VLC. VLC optimizes the codeword

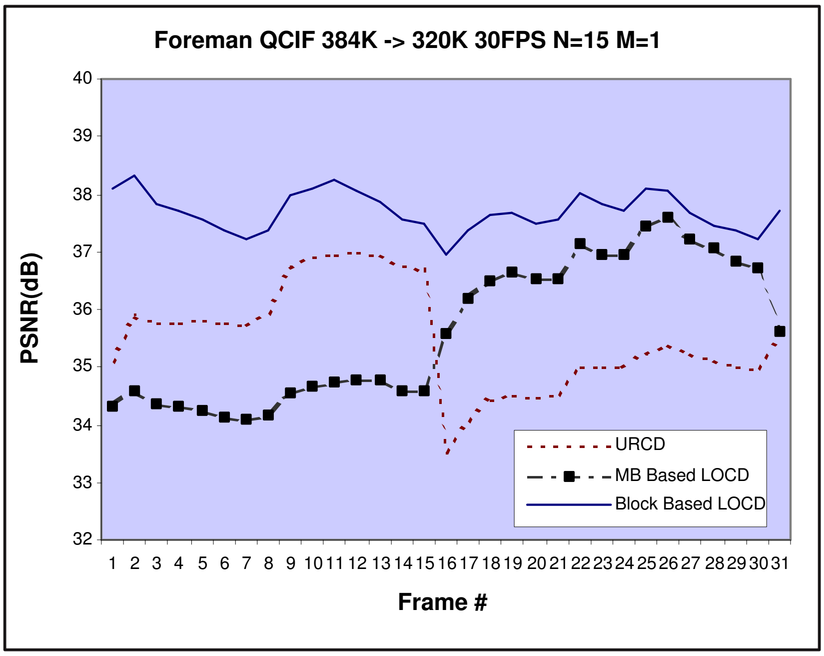


Figure 3.5: Frame-wise quality comparison for different CD approaches

length based on the symbol occurrence probability. When an artificial truncation point is set for a DCT block, the codeword lengths are no longer optimal and this the resulting bit stream length will be increased. Table 3.1.3.1 shows some examples of 3D-VLC coding results. Therefore we have:

$$\Delta b_{re} = b_{dct}^{k} - \tilde{b}_{dct}^{k} \tag{3.10}$$

where k is the truncation point, b_{dct}^{k} and $\tilde{b}_{dct}^{k}$ are the bit amounts of the k^{th} run length symbol before and after the truncation respectively. Please refer to Figure 3.2 (the row marked with star) for illustration.

3. The Lagrange optimization search sometimes fails to converge. The search of optimal Lagrange multiplier might not converge, consequently introducing bit

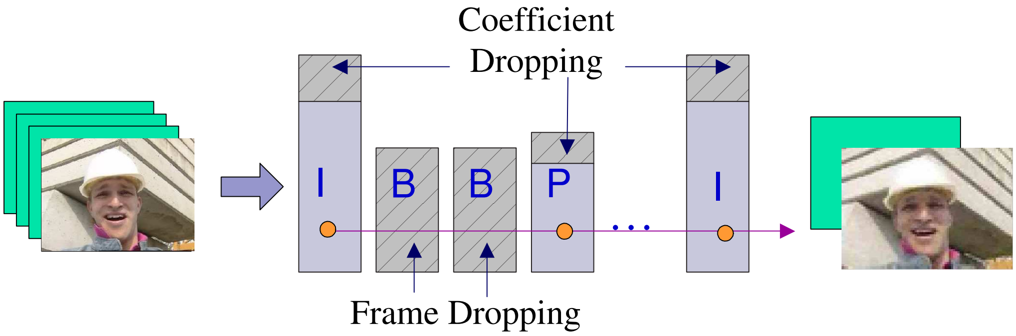


Figure 3.6: FD-CD combined MDA

drift:

$$\Delta b_{lo} = B_{keep} - \tilde{R}_k(\lambda) \tag{3.11}$$

where B_{keep} is the target bit rate for the frame and $\tilde{R}_k(\lambda)$ is the actual reached rate considering both the convergence failure and the truncation bit drift described in (1) and (2).

4. Some bits are padded at the end of each frame for alignment. The net effect is negligible because only up to 8 bits are involved per frame [78]. This padding effect will not be discussed.

Table 3.1: Some examples of codeword length change before/after CD

Level	Run	EOB	Coded	Bit Amount
-2	3	0	B0000100011	10
		1	B000001100011011	15
2	0	0	B01111	5
		1	B0000110010	10
-3	6	0	B0000001010100	13
		1	B00001111100011011111111111011	29

Among all of these reasons, Δb_{tr} and Δb_{lo} are associated with the URCD algorithm. Note $\Delta b_k = \Delta b_{tr} + \Delta b_{re}$, where Δb_k is the observed bit error after the truncation as defined in Equation (3.5). The bit allocation adjustment for URCD is simple: Δb_k will be merged into the next block. When the end of the frame is reached, Δb_k will be passed to the first block in next frame. E.g., the bit budget of next block will be adjusted as:

$$\Delta \hat{b}_{keep} = b_{keep} - \Delta b_k \tag{3.12}$$

For LOCD, Δb_{lo} is the observed bit drift for each frame. Since the bit budget allocation is optimized based on a frame-size window, the block-wise adjustment above employed in URCD is not suitable. Instead we propose to use the VLC re-encoded symbol to calculate the bit rate. I.e., Equation (3.7) will be rewritten as:

$$R_{\mathbf{k}} = \sum_{j=1}^{N} \left(\sum_{i=1}^{k_j - 1} b_{dct}^{j,i} + \tilde{b}_{dct}^{j,k_j} \right) \tag{3.13}$$

where $\tilde{b}_{dct}^{j,k_j}$ is the bit amount of VLC re-encoded symbol. This method is referred as VLC Re-encoding Drift Compensation (VRDC).

Furthermore, with the absence of FD, the observed drift in LOCD will be considered in next frame using way similar to that described in Equation (3.12), with the difference that frame bit budget B_{keep} is considered.

3.1.3.2 Rate Control for FD

The challenge of FD rate control is how to allocate bit budget for the remaining frames. In CD without FD, the bit budget is allocated to each frame with a fixed ratio. In FD, however, all bits of a frame are dropped and uniform distribution is not feasible. In order to keep a uniform quality among the kept frames, we adopt the GOP windowed rate control scheme typically used in video encoding rate control [39]. FD is more or less like the frame skipping case during encoding in order to avoid the buffer overflow. We proposed a rate control method to handle the FD-CD combined transcoding. The basic idea is to adjust the bit allocation for each processed frame dynamically through the adaptation process in a GOP. In the remaining of this section we will describe this method in details. First the definitions of the terms that will be used are present as follows.

FPS	Frame rate
N	GOP size
R	Original bit rate
R'	Target bit rate
B_f	Total bit amount in a frame
B_c	Bit amount for coefficient symbols in a frame
n_I, n_P, n_B	number of remaining I, P, B frames in a GOP without FD
n'_I, n'_P, n'_B	number of remaining I, P, B frames in a GOP with FD
w_I, w_P, w_B	Estimated weight for I, P, B frame in bit allocation
r_U	Estimated frame original bit amount unit
r'_U	Estimated frame target bit amount unit
$\tilde{\eta}_{bits}$	Estimated uniform truncation ratio

First, the bit budget for a GOP is allocated using the following calculation:

$$\text{Original bits for GOP: } R_{GOP} = \frac{R \cdot N}{FPS} \tag{3.14}$$

$$\text{Allocated target bits rate for GOP: } R'_{GOP} = \frac{R' \cdot N}{FPS} \tag{3.15}$$

Second, before transcoding, the estimated bit amount and the uniform truncation ratio are computed. In order to do this, we adopt w_I, w_P and w_B. Therefore,

$$r_U = \frac{R_{GOP}}{w_I n_I + w_P n_P + w_B n_B} \tag{3.16}$$

$$r'_U = \frac{R'_{GOP}}{w_I n'_I + w_P n'_P + w_B n'_B} \tag{3.17}$$

$$\tilde{\eta}_{bits} = \frac{r'_U}{r_U} \tag{3.18}$$

The truncation ratio $\tilde{\eta}_{bits}$ is used to guide CD operation. w_I, w_P and w_B control the influence on the bit rates in I, P and B frame. Empirically, the initialization values for them are set to be 4.0, 2.0 and 1.0 respectively. The precise initialization is not crucial because for different video streams they might vary, and during the adaptation, these weights will be adaptively updated according to the actually collected statistic, as will be introduced soon. n_I, n_P, n_B are decided by the GOP structure, while n'_I, n'_P, n'_B are decided by both the GOP structure and the FD mode. These six numbers will be adjusted after transcoding every frame. For a better understanding, Figure 3.7 illustrates the relationship among these parameters, where the first B frame in each sub-GOP is dropped The frames are aligned in the actual coding order.

Another important process is the weight adaptation. Throughout the transcoding, the weights w_I, w_P and w_B are updated according to the latest frame bit statistic. Denote b_I, b_P, b_B as the bit amount for the most recently observed I, P and B frames respectively. The updating process is done as follows:

$$w_B = 1.0 \tag{3.19}$$

$$w_P = \frac{b_P}{b_B} \tag{3.20}$$

$$w_I = \frac{b_I}{b_B} \tag{3.21}$$

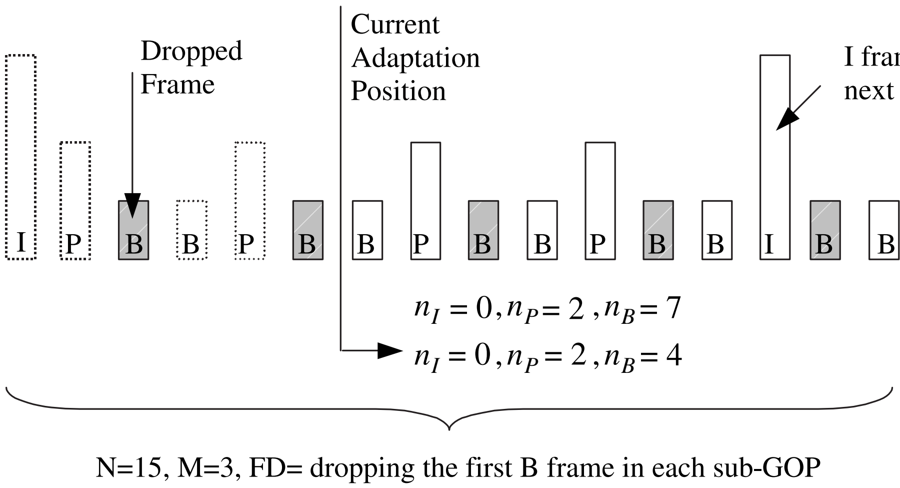


Figure 3.7: Parameters involved in FD-CD rate control

This FD-CD rate control method is called Adaptive Frame Bit Allocation (AFBA). AFBA is a very efficient way to handle the bit allocation issue for FD-CD adaptation.

3.1.3.3 Performance of rate control

Our FD-CD transcoder has been implemented in both the MoMuSys [50] and Microsoft [48] MPEG-4 verification model (VM) codes. Figure 3.8 shows the performance of our proposed rate control method when tested on the *Foreman* sequence. The transcoding operation changes the bitrate from $1.5Mbps$ to $800Kbps$. The actual bit amount for each GOP in the adapted video is shown in the figure. No adaptation means no VRDC is employed and a fixed bit impact weight ($w_I = 4.0, w_P = 2.0, w_B = 1.0$) is used. From the figure it is evident that both VRDC and AFBA have considerable contribution to the rate control. A combination of them can efficiently allocate bits uniformly under the bit constraint without severe vibration. The lower starting point of the VRDC+AFBA is due to the imprecise initialization value of the bit allocation impact weights. After first GOP, the weights are adjusted dynamically resulting in satisfactory rate control.

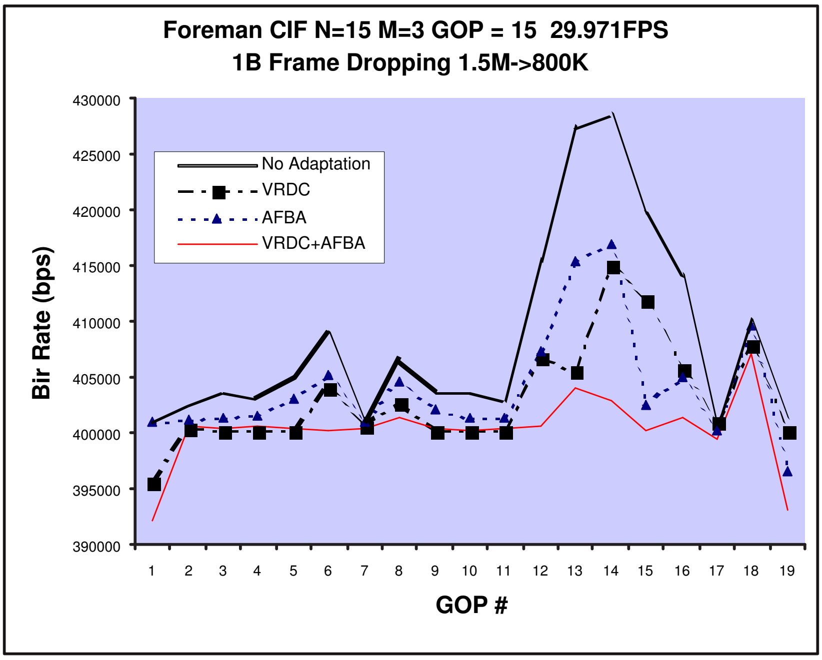


Figure 3.8: FD-CD rate control performance.

Figure 3.9 further shows the bit budget matching performance for the remaining frames in the first second of Foreman sequence. The matching error (in bits) indicates the difference between the allocated bit amount and the actual generated bit amount. Our proposed rate control method combing AFBA and VRDC accomplishes precise bit allocation and the error is much less than the one without rate control.

3.2 Utility Function based FD-CD MDA

In Section 2.2.3 we showed how UF can be formulated based on ARU space mapping. In a similar way UF can be constructed for FD-CD. Specifically, an FD-CD

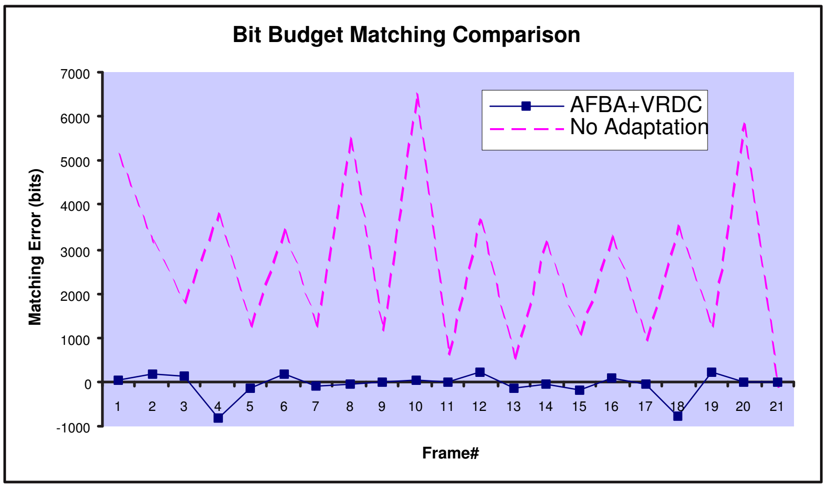


Figure 3.9: Bit budge matching performance

adaptation method can be described as $\mathbf{a} = (f, c)$, where f and c represent a specific frame dropping method and a coefficient dropping method respectively. We define the following operation for FD: "no frame dropping", "drop one B frame only", "drop all B frames", and "drop all B and P frames". CD is defined using a set of truncation ratio, i.e., $c\%$ of the bits. For instance, $\mathbf{a} =$ (all B-frames dropped, 10%) means all of the B frames in a GOP are dropped and 10% of the bits from each remaining frame will be reduced by CD. A typical UF constitutes a set of such FD-CD operation points, called anchor nodes, as shown in Figure 3.10. For a specific video clip, given an anchor node $\mathbf{a}$, its corresponding resource and utility value are denoted as r and u. Anchor nodes with the same FD are connected forming a curve, and the adaptation operations between two anchor nodes are obtained through linear interpolation. The whole set of the curves define the UF, which represents the utility-resource relation associated with the given video in response to the available adaptation operations (FD-CD).

To obtain a more efficient representation, we further simplify the representation

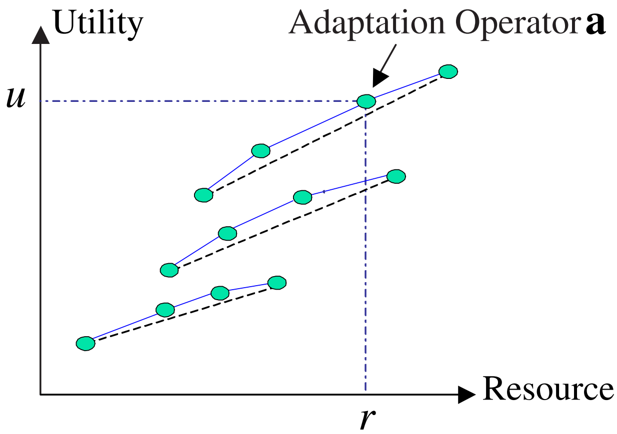


Figure 3.10: Definition of UF for FD-CD

of the UF by using the linear approximation of each curve as the dotted lines shown in Figure 3.10. The approximation is defined by two end nodes of each curve. Therefore the UF can be denoted as:

$$\mathbf{F}^{UF} = \left(f_1^{UF}, f_2^{UF}, ..., f_n^{UF}\right) = \left(r_1, r_2, ..., r_{2n}, u_1, u_2, ..., u_{2n}\right) \tag{3.22}$$

where each curve $f_i^{UF}, i = 1, 2, ..., n$ in Figure 3.10 is associated with 2 end points. Such approximation representation is very beneficial in reducing the dimensionality of the representation and improving the efficiency of the statistical prediction method described later. Our experiment demonstrates that such linear approximation provides a very satisfactory result in UF prediction (shown in Section 3.4.2). The ordering of the nodes does not matter, as long as a consistent scheme is maintained.

3.3 Content Based UF Prediction

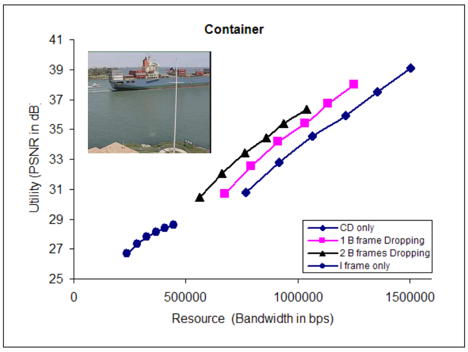
In section 2.3, we have analyzed and evaluated different approaches to approximate UF information, which is important for real time adaptation applications. Content based UF prediction framework is proposed. The usability of this framework can also be validated using the actual FD-CD UF shown in Figure 3.11. Figure 3.11(a) and (b) are the UFs for the 1^st and 2^nd second in the *Container* sequence respectively while Figure 3.11(c) and (d) for 1^st and 3^rd second in the $Stefan$ sequence respectively. It is very interesting to observe the relationships among different component curves of the UF's. The similarity between Figure 3.11(a) and (b) (or between Figure 3.11(c) and (d)) is obvious, such as the rankings of different curves at the same bitrate, the range of bitrate and utility of each curve, etc. On the other hand, the clips from different video sequences have quite different UF characteristics. This provides supporting rationale that video segments of similar content features will exhibit similar UF characteristics.

3.3.1 Problem description

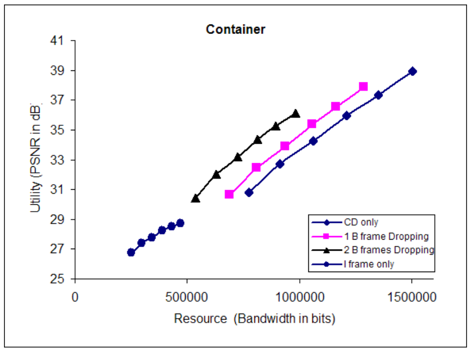
The issue of UF prediction can be formally formulated as follows: given the content feature $\mathbf{F^{CF}}$ of one video clip, develop a suitable function that maps the content feature space to the UF space, i.e.,

$$\mathbf{F^{UF}} = G\left(\mathbf{F^{CF}}\right) \tag{3.23}$$

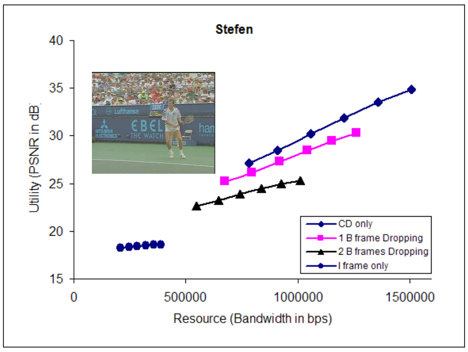
where $\mathbf{F}^{UF} = \left(f_1^{UF}, f_2^{UF}, ..., f_N^{UF}\right)$ is a N-dimension UF row vector and f_i^{UF} is i^{th} component of $\mathbf{F}^{UF}$, and similarly $\mathbf{F}^{CF} = \left(f_1^{CF}, f_2^{CF}, ..., f_M^{CF}\right)$ is the M-dimension content feature row vector and f_j^{CF} is the j^{th} component of $\mathbf{F}^{CF}$. Equation (3.23)



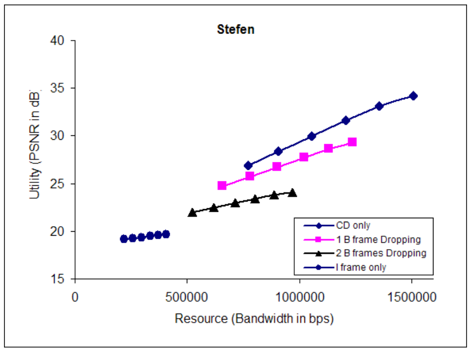
(a) Container 1st Frame



(b) Container 2nd Frame



(c) Stefan 1st Frame



(d) Stefan 3rd Frame

Figure 3.11: Relationship between video content and UF appearance

is a typical multivariate regression problem. For each f_i^{UF} in $\mathbf{F}^{UF}$, we want to find a mapping g_i, such that

$$\begin{aligned} f_i^{UF} &= g_i\Big(\mathbf{F}_{CF}\Big) = g_i\Big(f_1^{CF}, f_2^{CF}, ..., f_M^{UF}\Big) \\ G &= \Big(g_1, g_2, ..., g_N\Big) \end{aligned} \tag{3.24}$$

By using Taylor expansion, this mapping can be written as:

$$f_i^{UF} = g_i\Big(\mathbf{F}_0^{CF}\Big) + \nabla \mathbf{g}_i\Big(\mathbf{F}_0^{CF}\Big)\cdot\Big(\mathbf{F}^{CF} - \mathbf{F}_0^{CF}\Big) + O\Big(|\mathbf{F}^{CF} - \mathbf{F}_0^{CF}|^2\Big) \tag{3.25}$$

where $(\cdot)$ is the dot product of two vectors, and $\nabla \mathbf{g}_i\left(\mathbf{F}_0^{CF}\right)$ is the M-dimension partial differential row vector. By keeping the components in Equation(3.25) up to first order and ignore the higher orders, this mapping can be considered as a classic linear regression problem. Based on Equation(3.25), we can derive the following:

$$
\begin{aligned}
\mathbf{F}^{UF} &= \left(f_1^{UF}, f_2^{UF}, ..., f_N^{UF}\right) = \left[\begin{array}{cc} \mathbf{F}^{CF} & 1 \end{array}\right] \left[\begin{array}{cccc} \nabla \mathbf{g}_1^T & \mathbf{g}_2^T & \dots & \mathbf{g}_N^T \\ c_1 & c_2 & \dots & c_N \end{array}\right] \\
c_i &= g_i\left(\mathbf{F}_0^{CF}\right) - \nabla \mathbf{g}_i\left(\mathbf{F}_0^{CF}\right) \cdot \mathbf{F}_0^{CF} \\
\nabla \mathbf{g}_i^T &= \nabla \mathbf{g}_i^T\left(\mathbf{F}_0^{CF}\right), i = 1, 2, ..., N
\end{aligned} \tag{3.26}
$$

By applying the standard Least Mean Square Error (LSE) method, the optimal estimation of g_i, indicated as $\hat{\mathbf{g}}_i$, can be found to be:

$$
\hat{\mathbf{g}}_i = \left(\left(\tilde{\mathbf{F}}^{CF}\right)^T \tilde{\mathbf{F}}^{CF}\right)^{-1} \left(\tilde{\mathbf{F}}^{CF}\right)^T \tilde{\mathbf{F}}_i^{UF} \tag{3.27}
$$

where $\tilde{\mathbf{F}}^{CF}$ is the set of observed content feature vectors in the training data with each row corresponding to a training sample, and $\tilde{\mathbf{F}}^{UF}$ is the corresponding i^{th} component of observed UF. Moreover, the Taylor expansion works only in a small neighbor of the center $\mathbf{F}_0^{CF}$. Thus the first-order approximation is effective if the content feature space can be divided into some small areas, and the regression procedure is applied for each area separately. Specifically, this can be done by forming K such subareas $S_k, k = 1, 2, , K$, and conducting the above approximate estimation method for points within each subarea. Therefore, the problem can be modeled as a K-segment piecewise linear regression problem and the parameters (c_i and $\mathbf{g}_i$) can be obtained for each subset. In forming the partitions of the space, we can consider clustering in the CF space, clustering in the UF space, or a hybrid one that partition the CF space subject to some compactness constraints of corresponding UF values. In our work, in order to apply the local regression method discussed above we chose clustering in the CF space, plus combinations of classification techniques mapping content features to the CF clusters. Our experiment results presented later indeed confirm the superiority of this choice.

3.3.2 System architecture

The architecture of our proposed content-based prediction framework is shown in Figure 2.11. Details of each component will be described in the following subsections.

3.3.2.1 Content feature extraction

We adopt the content features based on the set adopted in our prior work [6] with minor modification. Three groups of features are considered: motion intensity, AC DCT energy, and quantization parameters. The first two groups of features embody the spatial texture complexity and temporal motion intensity information. The third group also indirectly reflects the scene complexity subject to the specific rate control algorithm used. They are extracted directly from the encoded stream or the stream metadata without decoding the video to the pixel domain. Our experimental results show that the performance of prediction can be improved if we also include the PSNR information from the metadata associated with the original encoded stream. Content features are extracted from each local video segment that is one second long. The length of the local segment is currently empirically determined, to keep an adequate balance between efficiency and accuracy. Note to ensure the video content in each segment is more or less consistent, we avoid shot boundaries within a segment by running automatic shot boundary detection and keeping the shot boundaries aligned with the segment boundaries. Although the shot boundary detection tool is not perfect, performance of the existing detection tools is quite high (precision up to 97% and recall up to 98% in [110]). Specifically, the following features are used in our system:

- Average motion intensity approximated by computing motion vector magnitude;
- Motion variance within the adaptation unit;
- Average percentage of macroblocks which have non-zero motion vector;
- Average I frame AC DCT coefficient energy;
- Average P frame AC DCT coefficient energy;
- Average quantization step size;
- Average PSNR if available in the stream metadata.

The average values are computed over the frames in the one-second segment. To further improve efficiency, we only process the I and P frames. The AC DCT energy of I and P frames are kept separate because our statistical feature analysis (Principal Component Analysis, PCA) shows they have distinctive contributions to the final performance. This is reasonable considering the DCT energy in the I frame is more related to the texture complexity due to the use of intra-frame coding, while the DCT energy in the P frames is mainly related to motion compensation residues because of the use of inter-frame coding.

3.3.2.2 Unsupervised clustering

The purpose of unsupervised clustering is to partition the content feature space into separate subspaces so that the regression technique can be applied in each local area. We adopt the K-Harmonic Mean (KHM) [108] clustering method, which in principle is related to the popular K-mean method. The main improvement of KHM over K-mean is by using the p^{th}-order harmonic distance, rather than the Euclidian distance. It was shown in [108] that KHM outperform K-mean in reducing the sensitivity to initialization and avoiding local optimal points.

Note the above clustering process is performed in the CF space, instead of the UF space. As shown in Figure 3.12, clusters formed in the CF space will ensure points in the same cluster have similar CF values. This is important for keeping a subarea of small variation of CF values and thus the first-order approximation by Taylor expansion described in Equation 3.25 remains valid. Although the alternative of doing clustering in the UF space can achieve compact data sets with similar UF values. The corresponding values in the CF space may be spread over a large range, and thus violate the assumption of proximity of the local regression method mentioned above. We will present performance comparison of these competing options later.

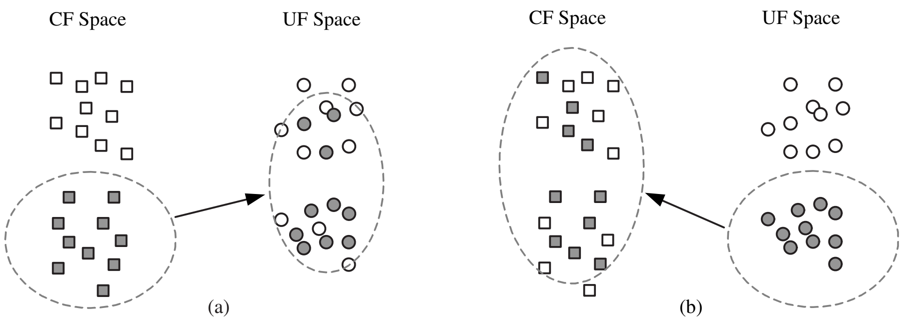


Figure 3.12: Difference between CF clustering and UF clustering. Shaded points show a cluster formed in one space and the corresponding values in the other space.

Another general problem of unsupervised clustering is determining the number of cluster, K. A K value that is too large will lose generality and result in overfitting, while a K value that is too small will result in significant bias. In our experiment, we determine the number of cluster through empirical trials and find $K = 16$ yields satisfactory performance. We expect the adequate choice of K depends on the characteristics of the video content and dynamic variations of the video over time. It is conceivable to propose some prediction schemes to determine the cluster number based on computable content features. Study of such methods and analysis of the effect on the UF prediction performance is beyond the scope of the current work.

3.3.2.3 Supervised classification by SVM

The purpose of classification is to categorize an incoming video clip into one of the classes and then apply the corresponding regression model to predict the UF. We employ Support Vector Machine (SVM) for the classification task. Basic SVM classifiers are for two-class discrimination. There are several ways to extend a binary classifier to support multiple-class separation, such as classifiers for one against others [88], or ones that fuse a set of two-class classifiers by methods like the Max Wins algorithm [32]. We adopt the directed acyclic graph SVM (DAGSVM) algorithm presented in [54] with minor modification to resolve the ambiguous region issue. In DAGSVM, the multiple-class classifier is constructed by using a Decision Directed Acyclic Graph. The classifier starts with separation between two most distinguishable classes using a regular two-class SVM. The negative class is excluded and the same two-class discrimination procedure is repeated for the remaining classes. It has been known to be a fast multi-class classifier with satisfactory performance [54].

3.4 Experiment Results

3.4.1 Experiment setup

In our experiment, we selected video from three movies to form the training and testing pools. The details of the video pool are summarized in Table 3.2. There were totally 2066 clips, each of which was one second long. The clips were carefully selected to cover a wide range of content features. Every clip was extracted from within a shot and thus no abrupt transitions like shot changes occurred within a clip. The proposed algorithm was tested using a standard cross validation procedure in which training and testing was done with random partitions of the pool (70% for training and 30% for testing) over multiple runs.

First, we need to compute UFs and extract content features for each set of training clips. In computing UFs, based on the given GOP structure ($N = 15, M = 3$), four FD operations were adopted: "no frame dropped", "the first B frame dropped in each sub GOP", "all B frames dropped", and "all B and P frames dropped". In the CD dimension, six CD levels were adopted: from 0% to 50% with 10% increment. As a result, there were totally 24 anchor nodes and four operation curves in each UF. Further details of the implementation are described in [94].

Table 3.2: Summary of data set

Video Source	*A Beautiful Mind* (736 clips), *Crouch Tiger Hidden Dragon* (589 clips), *Taxi II* (741 clips)
Clip Length	One second (2 GOPS)
Image Format	SIF (352×240) pixels
Video Compression	MPEG-4 with 30fps and TM5 rate control
GOP structure	GOP size N=15, sub-GOP size=3

Evaluation of the proposed prediction method can be based on various performance metrics. For example, errors in predicting the UF can be defined based on the L_2 metric as follows

$$D = \frac{1}{L}\sum_{l=1}^{L}\left\|\mathbf{F}_l^{UF} - \hat{\mathbf{F}}_l^{UF}\right\|^2 \tag{3.28}$$

where $\mathbf{F}_l^{UF}$ is the actual UF and $\hat{\mathbf{F}}_l^{UF}$ is the predicted one. L is the number of the test clips. Alternatively, the utility ranking of permissible operators at fixed bitrates can be evaluated, comparing results using the predicted UF verse the ground truth UF.

Table 3.3 is the specification of the algorithms employed in the experiment.

Table 3.3: Algorithm specification

Unsupervised Clustering	K-Harmonic Mean (KHM) using p^{th} order harmonic distance. $p = 0.5$, Number of clusters $K = 16$
Classification	DAGSVM multi-class classification: $C = 100$, kernel=RBF with $\gamma = 0.5$
Linear Regression	Trained by LSE algorithm. See Equation (3.24)
GOP structure	GOP size N=15, sub-GOP size=3

3.4.2 Performance

Figure 3.13 shows the prediction errors from four methods: our proposed method (Content Feature Clustering based Regression, CFCR); our proposed method but without local regression (Content Feature Clustering based Classification, CFCC), an alternative approach using clustering in the UF space instead of the CF space (UF-Clustering based Classification, UFCC), which is adopted in [6], and UF-Clustering based Regression (UFCR). The prediction error is measured by the L2 distance between the true UF and the predicted UF (see Equation (3.28)). The experiments were run for 10 times and the average performance was computed. The proposed method (CFCR) achieves the best result. That is to say when classification is combined with regression, clustering in the CF space is the best. This validates our decision in adopting the CF-space clustering method. However, it is interesting to note that without regression, techniques using clustering alone (UFCC and CFCC) favors clustering in the UF space. This is consistent with the UF-space clustering techniques used in the literature work [6]. Note in the pure clustering approach, the representative UF of each cluster as used as the predicted UF for all the points mapped to the same cluster.

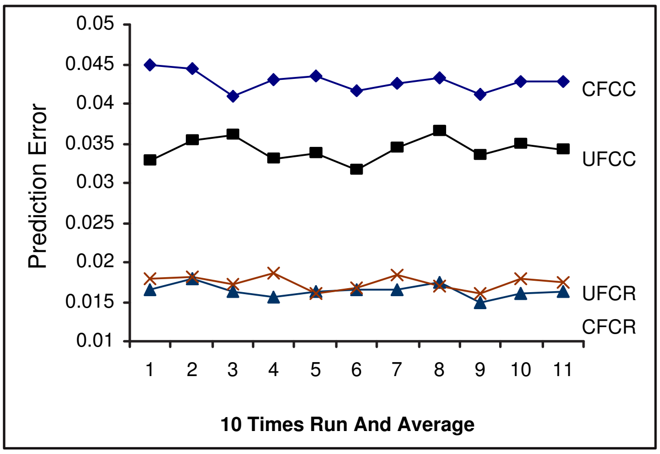


Figure 3.13: Comparison of the prediction performance in terms of prediction error.

Figure 3.14 further shows the performance improvement of CFCR over CFCC. 10% of samples from the testing pool are randomly selected and the results show that majority of the samples (86%) can benefit from regression.

Figure 3.15 shows comparison between some predicted UFs and the corresponding ground truth. The predicted UFs indeed match the true values very well. Typically, the prediction of the utility value (y axis) is not as good as the prediction of the resource value (x axis). However, the ranking of utility values among different transcoding options are quite consistent. Such ranking information provides the

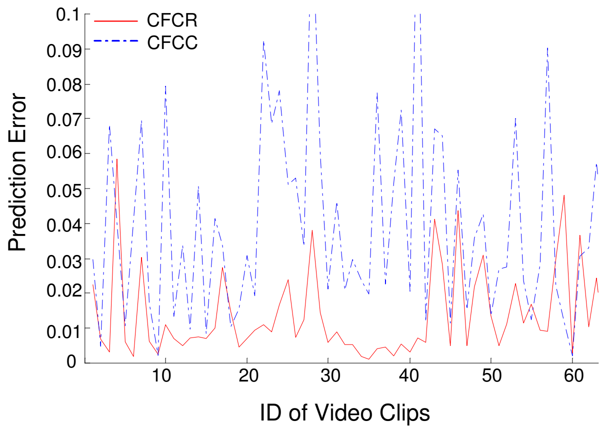


Figure 3.14: Performance improvement through regression.

most important input to our adaptation system for selecting the optimal transcoding option meeting a given target resource constraint.

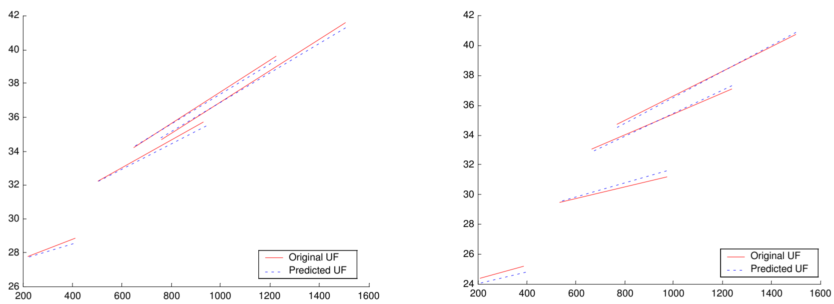


Figure 3.15: Matching predicted UF to the ground truths.

In addition, we also measured the accuracy in selecting the optimal operator given various target bit rates. Five typical bandwidths were used as the test target rates: $1.2M$, $1.0M$, $800K$, $480K$ and $320Kbps$. The original input video rate before transcoding was $1.5Mbps$. Our proposed method (CFCR) was compared with two alternatives: CFCC and the most frequent adaptation method. The latter did not take into account content features in each video, and simply selected the operation that achieves the highest quality for the most number of video clips in the training pool. Figure 3.16 shows our method outperforms the other two and exhibits significantly higher accuracy (up to 89%). From both the above evaluation criteria, the results are quite encouraging - the proposed content-based prediction method achieves very good accuracy in predicting the UF values as well as the ranking among competing adaptation operations.

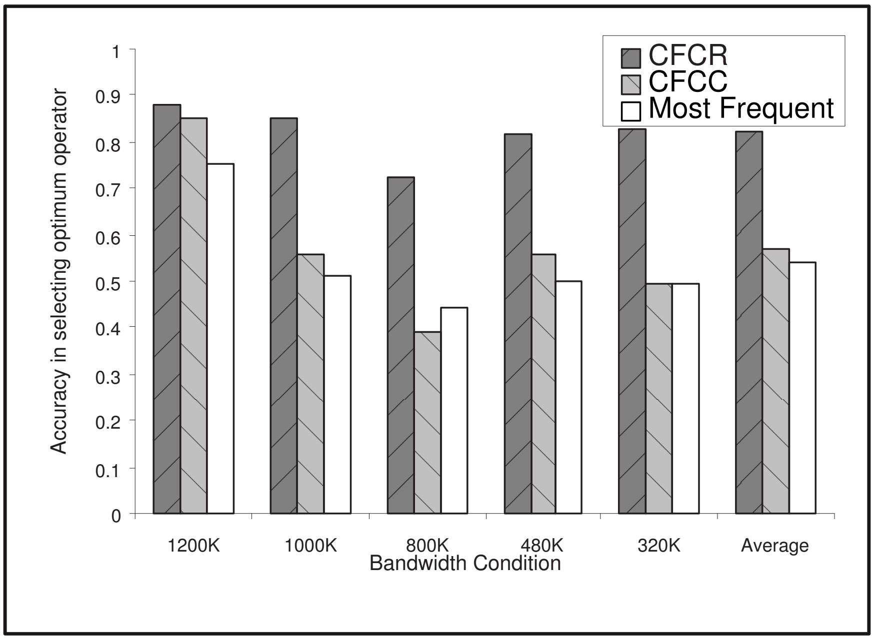


Figure 3.16: Performance in terms of prediction accuracy in choosing the optimal operator.

Besides prediction performance, computational complexity is another important factor for a real time application scenario. Because the MPEG-4 codec we used was not a real-time implementation, we were not able to provide the real time benchmark data. However, all of the computation processes in our system are light-weight. As shown in Figure 2.11 the main costs in our system include feature extraction and online prediction. The online prediction process, including classification and regression, can be implemented efficiently. Specifically, SVM classification only needs to calculate the kernel function and dot product between the content features and a sparse set of support vectors; linear regression involves only a multiplication between the model matrix and content feature vector. For feature extraction, partial bit stream decoding is necessary in order to obtain the content features, plus some minor extra calculation such as computing averages. The combination of all these computation steps is still much lighter than the complexity of a regular decoder (because the most complex component, motion compensation, is not needed). Considering video decoders can be implemented on most platforms with a real-time performance, it is reasonable to conjecture that our system can be implemented in a real-time fashion as well.

3.5 Summary

In this chapter we explore the SNR-temporal MDA behavior for MPEG-4 codec using FD-CD combined adaptation. The characteristics of FD and CD are analyzed, and the rate control issue involved in FD-CD is addressed explicitly.

The UF based MDA and content based UF prediction framework introduced in Chapter 2 are implemented and evaluated using MPEG-4 FD-CD adaptation operations. Specifically, we develop a formal UF representation based on the FD-CD adaptation operations. We further demonstrate the processes of discovering video classes trough clustering, developing classification schemes that automatically categorize new videos, and effectively predicts the UF using content features computed from videos. The experiment results demonstrate very promising performance - a high prediction accuracy (up to 89%) over diverse types of video content.

Chapter 4

Prediction of Preferred Multi-Dimensional Adaptation of Scalable Video Using Subjective Quality Evaluation

Scalable video coding offers a flexible representation for video adaptation in multiple dimensions thus providing great benefits for UMA applications. In this chapter, we apply the proposed utility function and content based prediction framework one of the state-of-the-art scalable video coding techniques, i.e., MC-3DSBC. We investigate the problem of predicting the optimal SNR-temporal adaptation operation combining SNR and temporal scalability. In contrast with the objective SNR metrics used in Chapter 3, we explicitly adopt evaluation metrics based on subjective video quality, called double stimulus impairment scale (DSIS) recommended by ITU-R [81]. We conduct extensive subjective experiment using a large pool of diverse video data (128 video clips) and a modest group of (31) human subjects. Formal statistical analysis is conducted to assess the statistical significance of the experiment. Based on the experiment data, we discover distinctive patterns of subjective preferences of different SNR-temporal resolutions when adapting video under different resource (bitrate) constraints. We observe that such distinctive patterns actually depend on the video content category - validating the assumption that video content is correlated with the video adaptation behaviors and thus can be used to predict the MDA operations in a classification-based scheme. The experimental results confirm the excellent accuracy in using compressed-domain visual features to predict the MDA operation matching subjective quality evaluation. In addition, we investigate the feature selection issue to identify the optimal set of content features that can be used to achieve the highest classification accuracy.

An important advantage of our solution is the flexibility of incorporating subjective quality evaluation with real-time online prediction. Subjective evaluation is done for the training data in the offline stage only. For a new video, the only computational processes needed are those for feature extraction and statistical classification, both of which do not involve heavy computations. Quality evaluations of different adaptation operations are not necessary for the new video, nor are the analytical models of the utility-resource functions like those used in the conventional R-D framework.

The rest of this chapter is organized as follows. Section 4.1 provides an overview on the MC-EZBC system. Section 4.2 introduces the scalability supported in MC-EZBC. In Section 4.3 the subjective experiment setup and result analysis are described, and the UF based on subjective evaluation is formulated. Section 4.4 presents the content-based prediction results. Section 4.5 summarizes this chapter.

4.1 Overview of MC-EZBC system

MC-EZBC coding systems are based on the 3D spatio-temporal decomposition of motion compensated video signals. The source video frames firstly undergo temporal decomposition. Figure 4.1(a) illustrates this procedure [30] where the GOP size is 8 and 3-level temporal decomposition is applied. The temporal filtering is applied along the motion trajectory. A pair of temporal low- and high-subband layers is generated for each two successive input frames, denoted as temporal layers. In the meantime one set of motion vectors is associated with these two frames and coded as the overhead. Temporal decomposition continues in an octave way until reaching the topmost level. For each temporal layer located in different hierarchical levels, spatial octave decomposition is utilized afterwards and the generated subbands are denoted as spatial subbands, which is similar with conventional wavelet decomposition employed in image coding systems, as indicated in Figure 4.1(b). Due to temporal filtering, each spatial subband contains interleaved spatio-temporal signals.

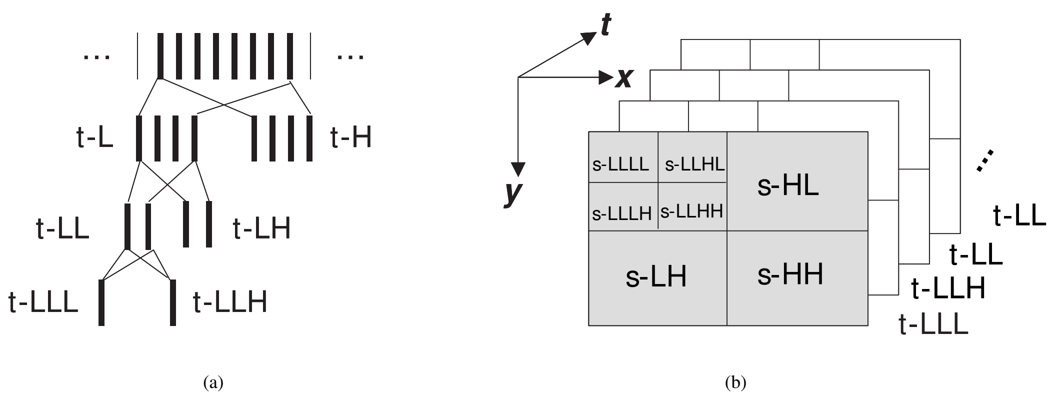


Figure 4.1: Spatial-temporal decomposition in MC-EZBC (a) Octave based five-band temporal decomposition; (b) The 3-D subband structure in a GOP.

After the decomposition, the transform coefficients will be further coded by the bitplane-based schema. There are several ways to construct the bitplane. Some typical examples in the literature include EZW [67], SPIHT [63] and SPECK [33]. These methods take advantage of the energy clustering of subband coefficients in space. A hierarchical set partitioning process is applied to split off significant coefficients (with respect to the threshold in current bitplane coding pass), while maintaining areas of insignificant coefficients in zero symbols. In this way, a large region of insignificant pixels can be coded into one symbol, thus providing efficient coding. The bitplane coding scheme adopted in MC-EZBC is the embedded zeroblock coding, whose mechanism can be explained as in Figure 4.2 [31]. EZBC is within the family of quadtree-based set partitioning coders. It contains two cascade procedures: quadtree buildup and quadtree splitting. In quadtree buildup, a hierarchical quadtree will be constracted, where each higher level node value equals to the maximum node value in the one level lower quadtree. This procedure is run from the pixel level until the whole processing block is merged into one node. Figure 4.2(a) shows such an example for 4×4 pixel block. In quadtree splitting, given certain quantization threshold corresponding bits (indicating whether the node is significant with respect to the threshold) are output. This procedure is done from the highest quadtree level until the pixel level is reached. Details about these procedures can be found in [31].

Each spatial subband undergoes the bitplane coding and the output bits are combined to generate an embedded bit stream. According to the requirement of scalability, the organization of coefficients from different spatial subbands in the bit stream might vary, depending on the scanning order of the subbands. One possible implementation is that the coefficients from both spatial subbands and temporal layers are fully interleaved. In each temporal layer, the scanning order is illustrated as track A in Figure 4.3(a). Before the process moves to next spatial subband, all of the coefficients in previous spatial subband from all temporal layers will be scanned first, shown as the track B in Figure 4.3(a). This will generate a fully interleaved bit stream of subband coefficients. The advantage of this method is its simple implementation, and spatial scalability is as easy as truncating and throwing

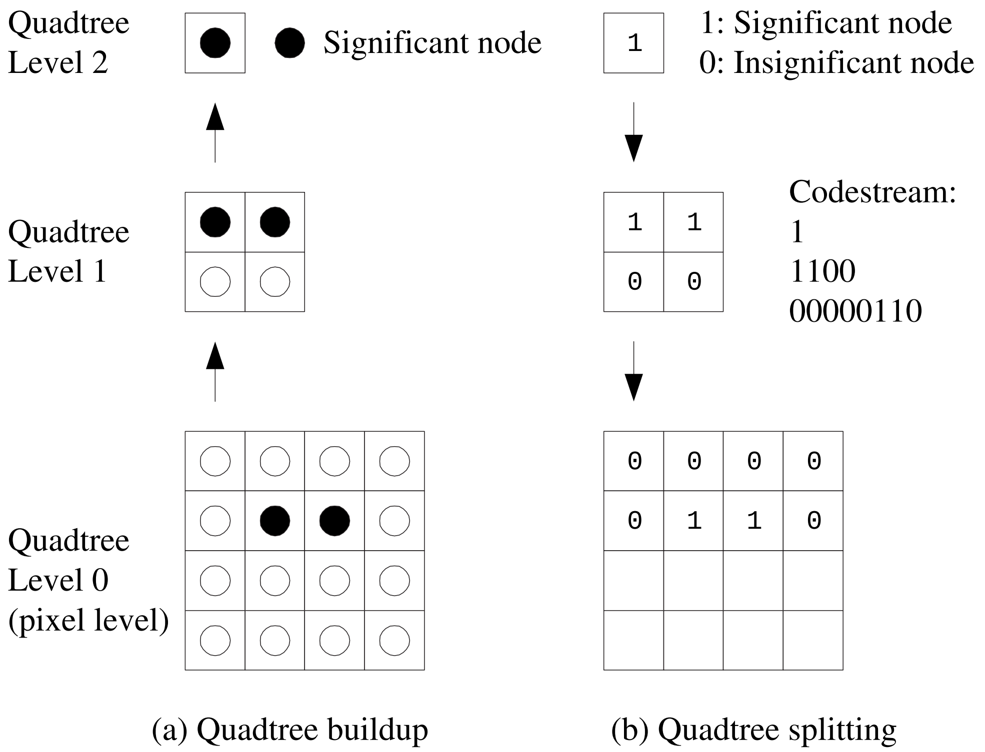


Figure 4.2: Illustration of quadtree construction and decomposition.

away the tail of the bit streams based on the bit rate limitation. The drawback of this method is also obvious: since the bits belonging to the same temporal layer are scattered at different locations in the stream, it is not convenient to realize the temporal scalability. Considering the shaded bit blocks in Figure 4.3(b), removing any of them need realign the whole bit stream, which is of low efficiency.

MC-EZBC applies an alternative way of coefficient organization as shown in Figure 4.4. First each temporal layer is clustered into several subgroups according to their temporal decomposition level. For each temporal subgroup, spatial subbands are further clustered in a similar way according to their spatial decomposition level. The coefficients located in a same spatial and temporal subgroup are coded into a fully interleaved substream. All of the substreams are aligned as shown in Fig-

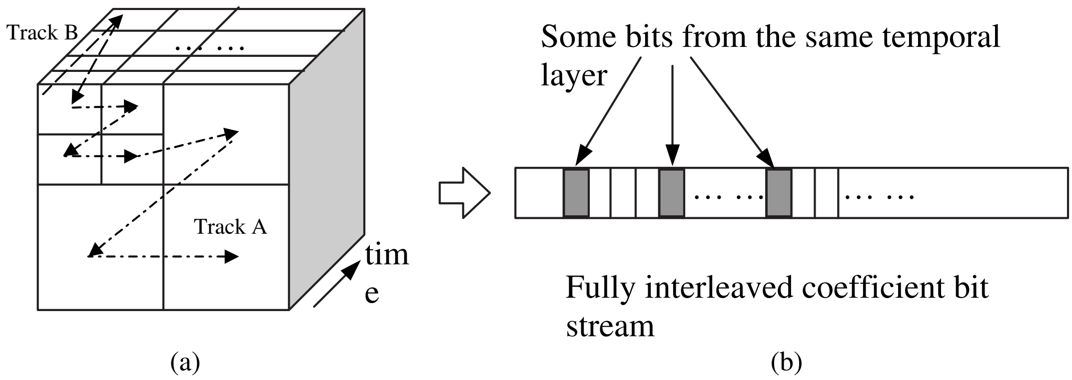


Figure 4.3: Fully interleaved bit stream generation.

ure 4.4. Note the temporal subgroup boundaries are determined after the spatial scalability operation is chosen, thus avoiding the scattering problem in Figure 4.3. This implementation is complex compared to the previous one. However, the advantage gained is its considerable freedom in terms of temporal scalability coding.

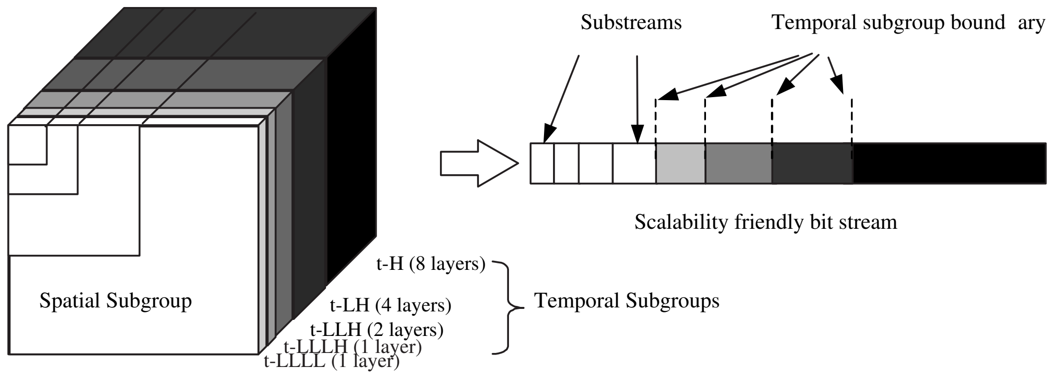


Figure 4.4: Scalability friendly bit stream generation.

4.2 Scalability in MC-EZBC

Compared with non-scalable video codecs, MC-EZBC provides much more efficient and flexible ways in supporting SNR-spatio-temporal adaptation. It does not requires modification or re-encoding of the coefficients and motion vectors. Instead, simple operations such as direct truncations of bitstreams are applied.

4.2.1 SNR scalability

SNR scalability in MC-EZBC is implemented through bitplane truncation, as shown in Figure 4.5, where only one temporal subgroup is considered. Within the temporal subgroup, as introduced in Section 4.2, the coefficients are coded into different substreams. All substreams are aligned based on their most significant bitplane (MSB). The truncation begins from the least significant bitplane (LSB). In Figure 4.5 each column represents the coefficients from the same spatio-temporal subgroup. The width of the column indicates the amount of coefficients. Theoretically, each substream has its own bitplane truncation boundary denoted as the bitplane index, say $K_{i,j}$, where i is the temporal subgroup index and j is the spatial subgroup index. The bit rate associated with $K_{i,j}$ is $R_{i,j} = R(K_{i,j})$. For a given vector $\vec{K} = (K_{0,0}, K_{0,1}, ..., K_{i,j}, ...)$, we can get a target rate $R_{i,j} = R(\vec{K})$. Obviously, there might be a set of different vectors $\mathbf{K} = \{\vec{K}\}$ reaching the same bit rate, SNR adaptation is responsible for selecting the optimal vector $\vec{K}_{op}$ that yields minimum quality distortion. Corresponding to the CD optimization discussed in Section 3.1.2, we have the following options: uniform truncation and Lagrange optimized truncation.

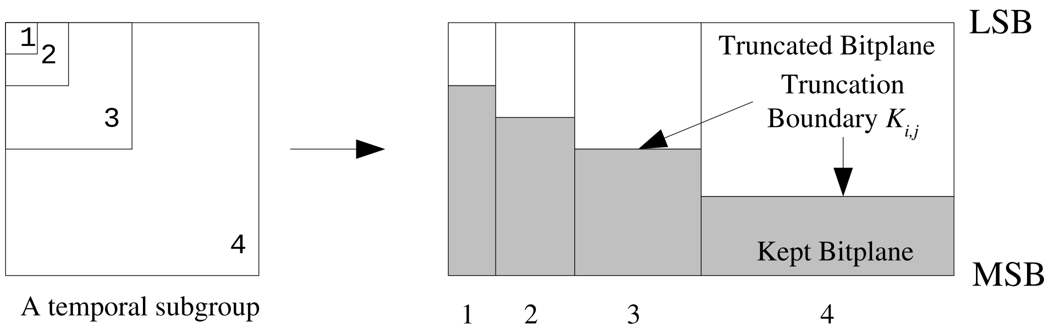


Figure 4.5: Bitplane truncation to realize the SNR-Scalability.

4.2.1.1 Uniform bitplane truncation (UBT)

As implied by its name, in UBT all of the substreams will have the same truncation boundary $K_{uniform}$. The term "uniform" is in the sense of bit plane amount, instead of bit amount. The reason for this is non-trivial. First, In MC-EZBC, the bitplane index is a natural indicator about the importance of the coefficients in terms of the perceptual quality. Second, due to the spatio-temporal decomposition structure, each subband layer has a different size (i.e., the amount of coefficients), as is indicated in Figure 4.5. Therefore, cutting the same number of bits from each layer may not produce an adequate perceptual quality. UBT is analogous to the uniform CD in Section 3.1.2.1. Despite of its simplicity, (as we shall see soon) UBT can achieve quite excellent SNR quality.

4.2.1.2 Lagrange optimized truncation (LOT)

Consider an arbitrary truncation boundary $\vec{K} = (K_{0,0}, K_{0,1}, ..., K_{i,j}, ...)$. The problem of optimal operation selection can be modelled as:

$$\tilde{\vec{K}} = \arg\min_{\vec{K}} \sum_{i,j} D\Big(K_{i,j}\Big) \tag{4.1}$$

$$s.t., \sum_{i,j} R\Big(K_{i,j}\Big) \leq R_0$$

where R_0 is the target bit rate. This problem can be solved by using Lagrange optimization method. The key issue is to find the appropriate mapping function $D(K_{i,j})$ and $D(K_{i,j})$. Conceptually the distortion can be modelled as:

$$D(K_{i,j}) = \sum_{P \in S_{i,j}} \Big(P - \hat{P}(K_{i,j})\Big)^2 \tag{4.2}$$

where $S_{i,j}$ is the substream defined by i, j, and $P, \hat{P}(K_{i,j})$ denote the original and the reconstructed coefficient respectively. Some existing embedded coding such as EBCOT [74] employed this model to select the optimal truncation point. One potential drawback is that it needs the computation of square values for each coefficient and is not convenient for bitplane-based embedded coding system. So we propose an approximate solution.

In EZBC, within each substream the coefficients are coded by using bitplane-based scanning. During each scanning, compared with a decaying threshold n, the coefficients are assigned into three lists: list of insignificant pixels (LIP), list of insignificant sets (LIS) and list of significant pixels (LSP) according to their magnitudes; and then some corresponding bits are output. The purpose of the coefficient scanning is to locate the significant coefficients while processing areas of insignificant coefficients into zero symbol (see Figure 4.2). For each given threshold n, the coefficients will undergo several rounds of scanning, each generating a sub bitplane with some corresponding output bits. The scanning in LIP and LIS might add some pixels into LSP, while the scanning in LSP refines the magnitude of the pixels in LSP. Figure 4.6 illustrates this procedure. Each sub bitplane will output some bits $r_{i,j,b,s}$ with corresponding distortion contribution $d_{i,j,b,s}$, where i, j denote the temporal subgroup index and spatial substream index respectively, and b, s denote bitplane index and sub bitplane index respectively. $r_{i,j,b,s}$ can be obtained from the bit stream. And $d_{i,j,b,s}$ can be modelled as a function of the effective number of pixels in LSP $m_{i,j,b,s}$. Specifically,

$$d_{i,j,b,s} = w_s \cdot m_{i,j,b,s} \cdot 2^{2b} \tag{4.3}$$

For LIP and LIS sub bitplanes, $m_{i,j,b,s}$ is defined as the number of pixels added into LSP after the scanning. For LSP, $m_{i,j,b,s}$ is defined as the number of pixels, which are affected by the refinement step and generate nonzero bits. w_s is a weight scalar related to the sub bitplane and defined by Islam, *et al* [33].

Given $r_{i,j,b,s}$ and $d_{i,j,b,s}$, the optimization problem is formulated as:

$$\begin{gathered} \min \sum_{i,j} \sum_{0 \le b \le K_{i,j}} \sum_{s} d_{i,j,b,s} \\ \sum_{i,j} \sum_{0 \le b \le K_{i,j}} \sum_{s} r_{i,j,b,s} \le R_0 \end{gathered} \tag{4.4}$$

By using the Lagrange multiplier method, we can find the optimal truncation boundary $\vec{K}_{op}$ for Equation (4.4).

Figure 4.7 shows the comparison between UBT and LOT. The rate operation is run for a target rate of $200Kbps$ on *Foreman* sequence with QCIF format, 1 second and 30fps. The PSNR of the target bit streams are 34.42dB and 34.23dB for UBT and LOT respectively. The curves show the ratio of bit allocation (i.e., the percentage of bits kept) for each substream. We can see that the ratios of bit allocation for both approaches are very similar. For the *Foreman* sequence, there are 5 temporal subgroups, each having 5 spatial substreams. Totally there are 25 substreams. In Figure 4.7 the substream ID is obtained by: SubstremID =

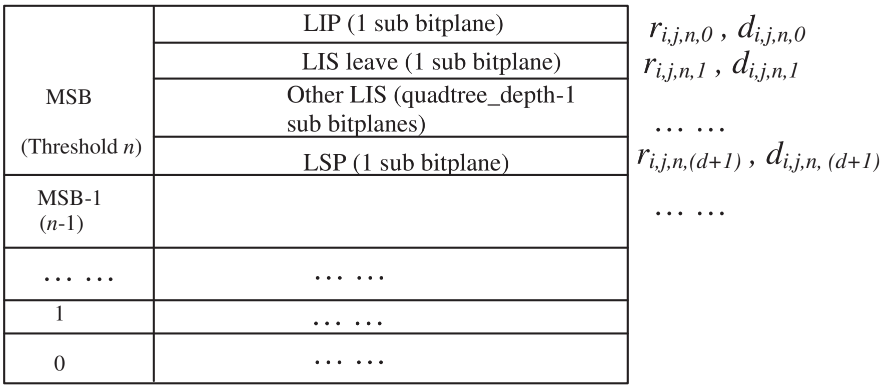


Figure 4.6: Bitplane scanning procedure in EZBC.

$5i + j$, where i, j are the index of the temporal subgroup and the spatial substream respectively. From the bit allocation ratio we can justify our assumption about the bit allocation policy: In order to get an optimal quality, (no matter spatial or temporal) low frequency components need be assigned more proportions of bits. It is interesting to observe that UBT, though extremely simple, can achieve excellent results. This observation is reasonable when we consider two factors of MC-EZBC: the embedded zeroblock coding method can effectively compress the coefficients based on their magnitude correlations and thus reduce the distortion dramatically; the context modelling can efficiently reduce the entropy of the bit stream and thus further improve the coding efficiency. In our experiments reported below, we will simply apply UBT.

4.2.2 Spatial and temporal scalability

In MC-EZBC spatial scalability is implemented by removing the whole substreams exceeding some specified spatial decomposition levels. Each spatial level dropping leads to a reduction of the image size by half in both width and height. Depending on the number of spatial decompositions, a different number of spatial scalability

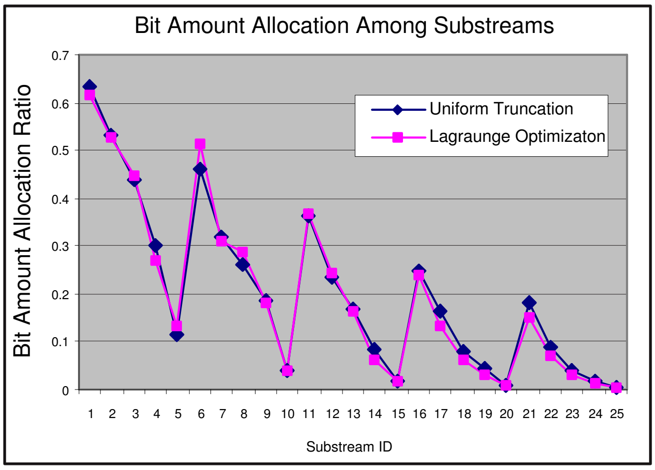


Figure 4.7: The ratio of bit allocation (the percentage of bits kept) for each substream.

levels can be obtained (e.g. QCIF, CIF, 4CIF etc.). Spatial scalability leads to different visual performances at various bit-rates and temporal resolutions and the quality of the adapted video is related to the specific implementation. For instance, the $2D+t$ methods [1] have been shown to provide improved performance at lower spatial resolutions as opposed to $t+2D$ solutions. Nevertheless, in MC-EZBC, our subjective experiment [96] showed that SNR-temporal adaptation always outperformed SNR-spatio-temporal adaptation in a wide range of bandwidth no matter using objective or subjective quality evaluation. Therefore we will not adopt spatial scalability in forming the MDA operations.

Similarly, temporal Scalability is realized by removing the higher frequency temporal coefficients and reducing the temporal resolution of the transmitted bit stream. Temporal scalability of MC-3DSBC codecs may result in an improved performance as opposed to conventional video coding schemes due to the long low-pass filtering process that separates noise and sampling artifacts.

4.2.3 SNR-temporal MDA in MC-EZBC

The SNR and temporal adaptations can be denoted as $\mathbf{a} = (a_{SNR}, a_{temp})$. Usually there are a set of different adaptation operations satisfying a given target bit rate R_0, while yielding different perceptual quality. Specifically, given a target adaptation bit rate, this allows two dimensions of adaptation to meet the bitrate - truncating the bit planes of the spatial subbands or changing the temporal resolution. The choices of the adaptation in the temporal dimension are discrete, and the choices of the bit plane truncation are more fine-grained. This case is exactly an instance of the M-D scenario depicted in Figure 2.6(a). Given the bitrate constraint, candidate points satisfying the constraint are reduced to a subset in the 2-D SNR-temporal adaptation space. The subset of points can be conveniently indexed by discrete labels in one of the adaptation dimensions. In the subsequent parts of the paper, we will index each SNR-temporal adaptation point by referring to its corresponding frame rate (i.e., full, half, quarter frame rate). Rate control will not be an issue any more because bitplane truncation in SNR scalability guarantees bit-wise precision automatically.

To choose the best SNR-temporal operation, a quality evaluation metric need to be assigned as the utility. In Chapter 3 we adopted PSNR as the utility value. One drawback of this scheme is that PSNR has difficulty in matching perceptual subjective quality especially for SNR-temporal combined adaptation. For a better understand, Figure 4.8 provides a visual illustration on this problem, where the *Foreman* sequence was coded at $100Kbps$ with different frame rate. The PSNR value averaged over the sequence gives a misleading measurement, in contrast with the degradation mean opinion score (DMOS) obtained by subjective perceptual quality evaluation. On the other hand, though in the literature many analytical models were proposed trying to simulate human vision system [16, 47, 62, 87, 97] in evaluating video quality, these models either need further justification, or have difficulty in estimating video quality when different SNR-temporal adaptation operatinos are involved. Therefore, in our work we directly adopt the subjective evaluation as the utility metric. In the next section, we will specify the subjective experiment, from where the utility function will be formulated.

(a) 15fps, 100Kbps, Average PSNR=25.98dB, DMOS=1.5/5

(b) 30fps, 100Kbps, Average PSNR=24.56dB, DMOS=2.2/5

Figure 4.8: Mismatch between subjective evaluation and MSE based measurement.

4.3 Subjective Quality Evaluation of SNR-Temporal Adapted Videos

As discussed above, signal distortion such as SNR is not adequate for comparing the video quality of different temporal rates. Instead, subjective quality evaluation is a more suitable measurement for SNR-temporal adated video sequences. Recent work in [102] included subjective experiments and stated that generally 15 frames per second was preferred for low-bit-rate coding. An explanation based on motion behavior in the video was presented. However, thorough studies of the influence of other content features like spatial complexity on the frame rate preference were not conducted. In addition, the experiments were done for the source coding process, instead of the adaptation of existing coded videos. This may influence the behaviors of the videos and their quality under different spatio-temporal configuration. The work presented in Chapter 3 has indicated significant correlation between the spatial complexity (texture) and the spatio-temporal rate preference. In another prior work [58], the preferred adaptation frame rate of MPEG-4 FGS video was studied based on some limited subjective studies. The preliminary data indicated that the subjects would prefer more spatial details when the spatial quality (e.g., PSNR values of the frames) was below some threshold. Once the spatial quality exceeded the threshold, smoother motion perception, i.e., higher frame rate, was preferred. However, [58] did not include sufficient experiment data to prove statistical significance and the quantitative analysis of the threshold values were not investigated. Therefore, we launch an extensive subjective experiment aiming at an in-depth understanding of the inter-dependence of optimal adaptation operation and user, bandwidth, and video content characteristics. In the following subsections, we describe the process for constructing the video pool, the subjective evaluation metrics, the analysis of user behavior consistence, statistic significance analysis of the data, and finally the clustering process to group videos with distinctive adaptation behaviors.

4.3.1 Video pool construction

The video pool used in our experiment consisted of three parts: standard test sequences such as *Akiyo*, *Foreman*, *Paris* and *Mobile*; test sequences used by Video Quality Experts Group (VQEG) during their video quality modeling evaluation work [62]; clips taken from commercial movies[1]. There were 128 video clips in total. All clips were 288-frame long[2], with CIF(352×288) resolution and original frame rate of 30fps. They covered a wide range of content characteristics and thus were suitable for our goal in studying the effect of content on adaptation behavior. Also, to ensure that each clip has consistent content features, we made certain that no shot boundary existed in the middle of each clip. Although the video content may still vary within a clip, most content features were consistent in such short clips. All of the clips were coded using the MC-EZBC codec [9] with the GOP size of 16 frames. The chosen coding parameters are summarized in Table 4.1. The bit rates used in the experiment were $50, 100, 200, 400, 600, 1000Kbps$, covering a wide range of bandwidth frequently seen in practical applications, with an emphasis on the low bandwidth end. Each clip was adapted into versions of all different bandwidths whenever the bit rate is achievable by the codec.

Table 4.1: MC-EZBC coding specification

GOP Size	16 frames
Decomposition	5 temporal levels, 6 spatial levels
Temporal Filter	2-tap Haar filter
Spatial Filter	Daubechies 9/7 filter
Motion Compensation	Bi-directional Variable Size Block Matching
Bitplane Coding	Zeroblock coding with context modelling

4.3.2 Subjective Experiment

The subjective experiment was carried out in a quiet, separated conference room. The video clips were displayed in a 19-inch Dell P991 Trinitron monitor at a resolution of 1280×960. The viewing distance was fixed at five times of the picture width.

[1] For movie clips, YUV sequences are extracted from the encoded bitstream that is coded around 2Mbps.

[2] Some clips from VQEG could only provide 240 frames (15 GOPs).

31 subjects in total participated in the experiment. They were undergraduate and gradate students at Columbia University from different departments. Due to the volume of the evaluation, the video pool was divided into 8 groups, each with 16 distinct clips. Each video group was assessed by 5 subjects. Some subjects enrolled in more than one group. We adopted the double stimulus impairment scale (DSIS) experiment recommended by the ITU-R standard [81] with minor revision. The experiment panel is shown in Figure 4.9. Four display windows were aligned in two rows and two columns. The left-top window displayed the un-adapted reference sequence. The other three windows displayed the adapted clips. According to the MC-EZBC architecture, these three clips were adapted into the same bandwidth with full frame rate (30fps, without temporal adaptation), half frame rate (15fps) and quarter frame rate (7.5fps) respectively. Their display windows were randomized to avoid opinion bias. During the experiment, the reference clip was firstly played. When it was finished, the user could choose to see the adapted clips one by one and give a Degradation Mean Opinion Score (DMOS) ranging from 1 to 5, corresponding to the worst quality to the best quality based on the perceived impairment when comparing the adapted video with the reference sequence. The subjects had the freedom to replay any clip if necessary. For each clip, all of the adapted versions at different bandwidths were evaluated resulting in a score for each (clip, bandwidth) combination. The temporal order of evaluating different (clip, bandwidth) pairs was randomized to avoid any potential bias.

4.3.3 User behavior consistency

In addition to the 128 test clips, we included 3 baseline clips that were seen by all 31 subjects. We used these three clips to assess the consistency of preferences among users. The 3 common clips were of diverse content characteristics. Each clip was

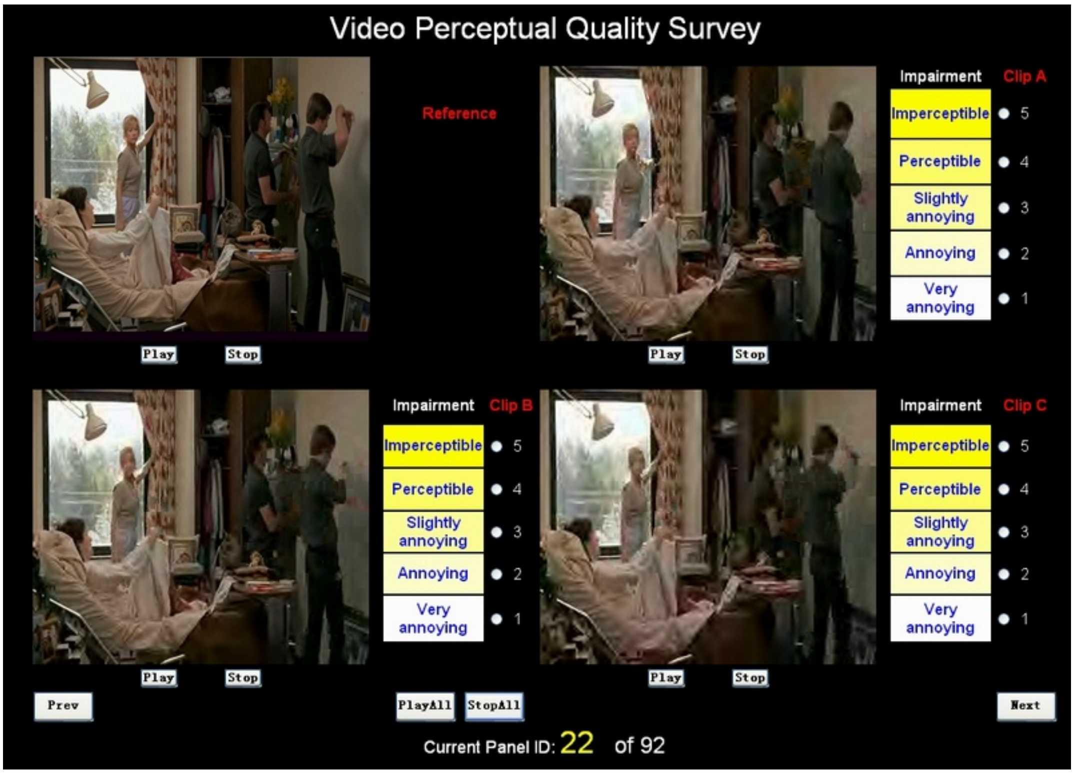


Figure 4.9: Subjective experiment panel based on ITU-R DSIS.

tested at 6 different bandwidths. For each video-bandwidth pair, each user assigned subjective scores of different temporal rates - resulting in an 18-dimensional score vector for each user over the baseline video set. The correlation matrix of the score vector for all users was calculated and is shown in Figure 4.10, in which users were re-sorted based on their mutual similarity. From the figure we can see that most of users (within the dashed region) behaved similarly with high or medium correlation, with others (about 5 users) behaving in a relatively dissimilar way. In other words, this indicates there is a high degree of agreement among preferences by a great majority of users.

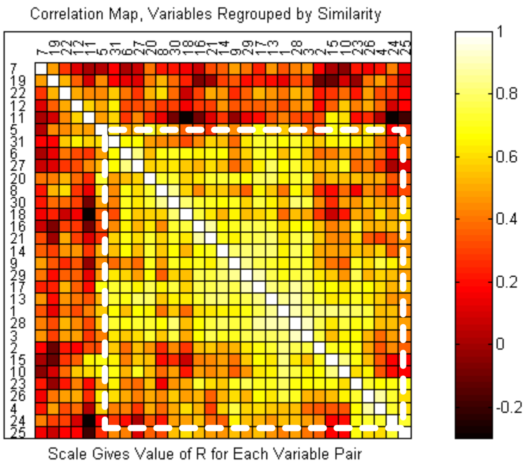


Figure 4.10: Correlation matrix for assessing user behavior consistence.

4.3.4 Statistical data analysis

For each clip-bandwidth pair, the experiment produced three different adapted versions (full frame rate, half frame rate, and quarter frame rate), each of which received DMOS scores from 5 different subjects. The next step was to apply statistical analysis to assess the ranking among different adaptations and the statistical significance of such rankings.

Firstly, the mean scores for three adaptation methods were ranked for the given (clip-bandwidth) pair, resulting in a descending ranked list of adaptations a_1, a_2, a_3, where a_i is the adaptation that has the i^{th} rank subjective score. Then, we used a paired t-test [45] technique to calculate the confidence score $P_{i,j}$ for the claim that adaptation a_i is preferred to a_j. Given a confidence threshold P_η, the following rule was employed to resolve any inconsistence among the pair-wise preference relations among all three adaptations:

if $P_{1,2} \geq P_\eta$ and ($P_{1,3} \geq P_\eta$ or $P_{2,3} \geq P_\eta$)

a_1 is the optimal operation;

else if $P_{1,2} < P_\eta$ and $P_{2,3} \geq P_\eta$)

a_1, a_2 tie and are the optimal operations;

else all other cases

three of the operations tie;

The threshold P_η plays an important role in determining the significance of claims about adaptation preferences. The higher threshold value we set, the more confident we can be in claiming about preferences of specific adaptations. However, since the experiment data is not unlimited, a higher confidence threshold also results in more cases of ambiguity, namely, "tie" as defined in the above procedure. Therefore, we need to find a balance between a high confidence and a low number of ties. Figure 4.11 illustrates the relationship between the number of ties and the value of P_η. We select $P_\eta = 0.75$ for our experiment, leading to a moderate amount of ties (around 25%). Although the selection is ad hoc and the data size (five subjects per clip-bandwidth pair) may appear to be limited, we will confirm the effectiveness of such experimental approaches through a satisfactory accuracy in adaptation prediction later.

4.3.5 Utility Function

Figure 4.12 shows the utility function generated based on the statistical analysis introduced above. Each connected curve depicts the adaptation results using the same temporal adaptation operation. On each target adaptation bandwidth, the vertical bar represents the opinion variance among the subjects. This subjective UF can be used the same way as introduced in Section 2.6. Nevertheless, we propose to use an alternative approach to generate UF information based on the behavior of a group of video clips. This approach greatly simplifies the implementation of

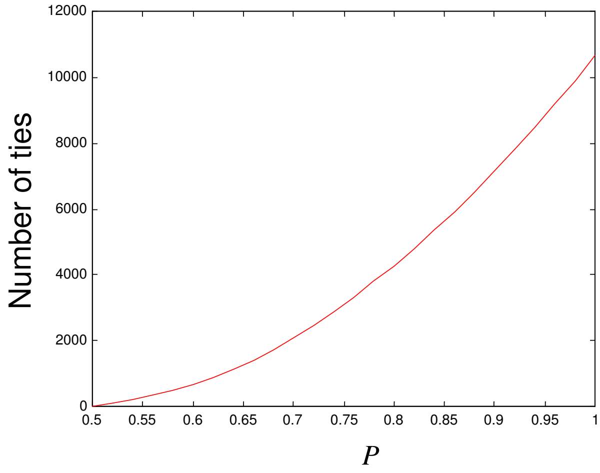


Figure 4.11: Number of ties v.s. P_η.

content-based prediction and maintains excellent prediction performance.

4.3.6 Prediction of optimal adaptation operation

Figure 4.13 summarizes the histogram of subjective preference for different frame rates at different bandwidths over the whole video pool. This histogram was generated by counting the number of subjects preferring each different adaptation operation at specific bandwidths. In the case of tie, the counts were split equally to tied adaptations. From the histogram, it is clear to see the trend: from high, medium to low bandwidth, the preferred adaptation operation shifts from full frame rate, half frame rate to quarter frame rate gradually. Such a trend is intuitive and re-confirms earlier findings [58].

The statistics shown in Figure 4.13 also reveals a very important piece of information - there exist distinct switching bandwidths r_{s_1}, r_{s_2} at about $200Kbps$ and $450Kbps$ at which the preferred frame rate changes. The preferred frame rate for video adapted above r_{s_2} is 30 frames per second. Half frame rate (15 fps) was pre-

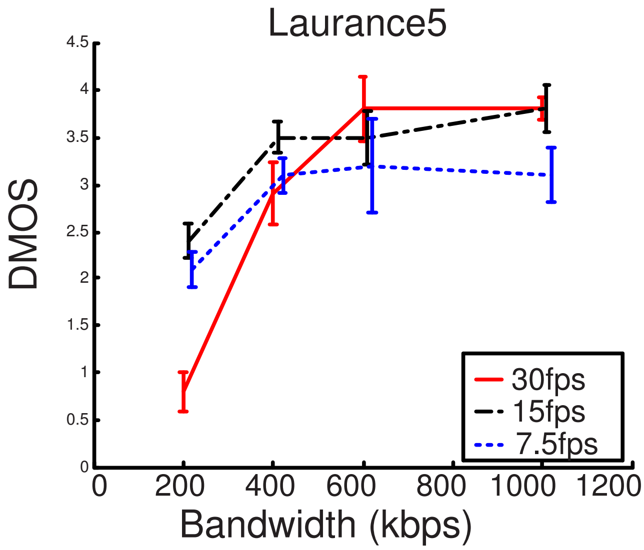


Figure 4.12: Utility function using subjective quality evaluation.

ferred when the bandwidth is below r_{s_2} but above r_{s_1}. When the bandwidth drops below r_{s_1}, the preferred frame rate becomes quarter rate; i.e., 7.5 fps. Due to the limited data sampling, it is hard to pinpoint the exact values of the switching bandwidths, but we can still infer a reasonable range (e.g., r_{s_2} is located in the range of $[400, 500]Kbps$). The above finding has very useful impact on practical applications - it provides a coarse prediction of the adaptation operation (with different frame rates in this case) at any given bandwidth. Such information can be used as UF to facilitate the selection of MDA operation.

4.3.7 Video clustering

The classification-based prediction paradigm requires solutions of two problems: construction of video classes and methods for mapping new videos to the classes. The latter will be discussed in Section 4.4. The former is a fundamental issue for our framework. And there are two basic approaches: manual construction based on

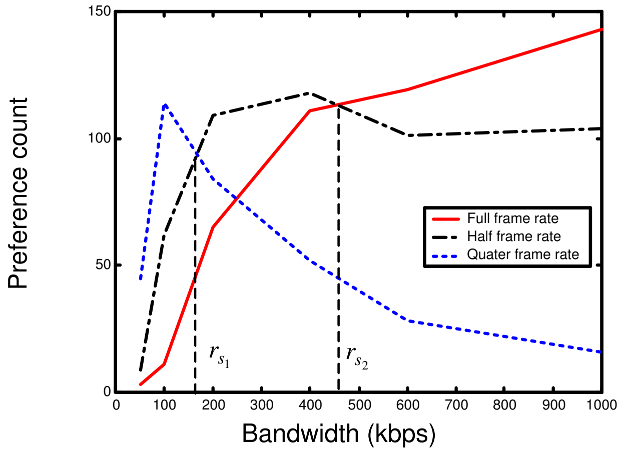


Figure 4.13: Histogram of the operation preference.

domain-specific knowledge, and full automatic discover by unsupervised clustering. The first approach utilizes prior knowledge (if available) about the domain to help define the classes. For example, for MC-EZBC we are using, we observe that the Minimal Achievable Bandwidth r_{MAB} of each video stream is a good indicator of the video content complexity and thus can be used as good criteria for defining video categories. r_{MAB} is defined as the minimal bandwidth achievable by any possible adaptation. It corresponds naturally to the intrinsic spatio complexity and motion activity of the clip. In our experiment, r_{MAB} had three distinct values: 50, 100 and $200Kbps$. Therefore, the clips in the video pool were labeled using three corresponding categories with low, medium and high content complexity respectively. Each category of videos has its unique patterns of adaptation preference, as illustrated by the preference histograms in Figure 4.14. We can see a clear trend that when the video content complexity increases, the switching bandwidths also shift to the higher end. This is quite intuitive: more complex videos need more bits for spatial details before a higher frame rate is needed. The above finding is significant, laying the foundation for predicting the optimal MDA operation based on the content features.

r_{MAB} is a good domain-specific knowledge that can be used to categorize the videos, and it can be easily obtained from parsing the MC-EZBC bitstream. However, such knowledge depends on the coding mechanisms and is not always reliable for generic codec such as MPEG-4 discussed in Chapter 3. In this situation our prediction framework utilizes automatic unsupervised clustering to discover video categories without any user supervision. To evaluate the usability of such method, herein we also apply an entropy-based unsupervised clustering method, called COOLCAT [2], to discover the video clusters. The cluster number was set to 6 empirically based on the cross-validation criterion. The clustering process was carried out in a feature space, in which the adaptation behavior of each video is represented by a set of features including the preferred frame rates at different bandwidths. Due to the space limit, readers are referred to our prior work in video category discovery [95] and the statistical clustering tool [2] for the detailed processes used in unsupervised clustering. The performance of such unsupervised clustering techniques, however, are compared against other alternatives in Section 4.4.

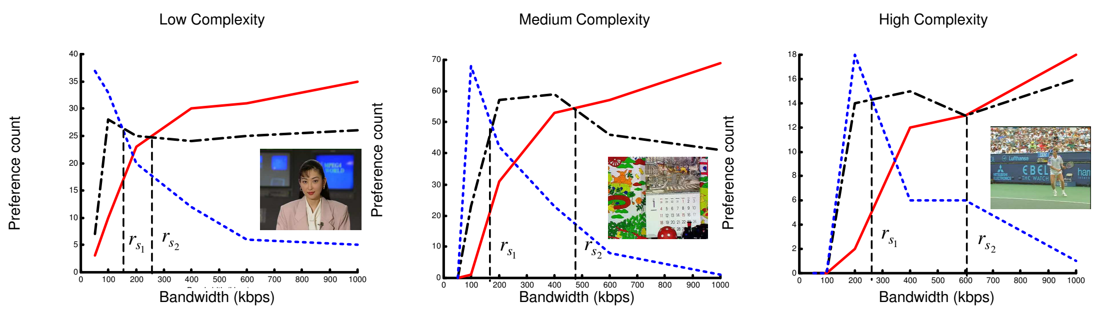


Figure 4.14: Histogram of preferred frame rate for videos with different content complexity.

4.4 Classification based prediction of optimal adaptation operation

Using the subjective quality evaluation data obtained through the extensive experiments, we next apply pattern classification techniques to develop classifiers and predict optimal MDA operation in a real time scenario as introduced in Section 2.3. The classification problem is formulated as the following. Given the features extracted from each input video, classify the video to one of the classes defined in the previous subsections. Following the discussion in Section 4.3, although classes defined upon r_{MAB} can be directly achieved, such direct class information may not always be available during real time processing. Therefore, we also demonstrate the feasibility of using content features to achieve the classification.

4.4.1 Content feature selection

During the machine learning and classification processes, content feature selection is an important step. Since the video adaptation scenario considers only pre-encoded video as input, we focused on the features that can be readily extracted from the MC-EZBC encoded format. Specifically our raw content feature candidates consisted of about 1200 variables, including the block size, block type, motion magnitude, motion phase, residual energies, etc., each computed from multiple frames within the clip due to the spatial-temporal interleaving used in MC-EZBC. It is not surprising to confirm through simulations that adoption of the whole feature set without selection indeed results in poor prediction accuracy because adding noisy irrelevant features usually hurts the classification performance. Based on HVS mechanisms discussed in Section 2.3.2.1, we include the features related to the spatial texture complexity and the temporal motion intensity. These two attributes reflect the spatial and temporal frequency details, which shape the contrast sensitivity functions (CSF) of HVS. For MC-EZBC, the most important set of features are related to three components: 1) the variable block size distribution. MC-EZBC employs hierarchical variable block size matching during the motion estimation, and therefore the variable block size is well associated with spatio-temporal details. 2) motion magnitude, indicating the motion intensity between two temporal frames. 3) residual energies, indicating the squared coefficient magnitude after the motion-compensated spatio-temporal decomposition. Therefore, the following two steps were applied during feature selection. First, only the features belonging to the three categories above were kept and the remaining excluded. Furthermore, the kept content features were merged by summing up the ones within the same category and the same spatio-temporal subbands. After this step, 86 combined features were kept. Specifically, these features were 1) 20 variables describing the block size histogram; 2) 36 variables describing the motion magnitude histogram; 3) 30 variables describing the residual energies. In order to further simplify the classification process, we applied mutual information feature selection (MIFS [3]) method to select a subset of features from the above 86 features. Specifically, if we want to select K features from the original feature space F_0, the following MIFS procedure was conducted to select one feature, f_k, in each iteration (k).

$$f_k = \arg\max_{f \in F_k} \{I(C, f) - \beta \sum_{s \in S_k} I(f, s)\} \tag{4.5}$$

where C is the video class labels, F_k and S_k are the remaining feature set and selected feature set at iteration k respectively (note $F_k + S_k = F_0$), I is the mutual information (MI) between two random variables, and β is a weight to regulate the relative importance of the MI between the candidate feature and the selected features with respect to the MI between the candidate feature and the output class. The higher β is, the more the algorithm penalizes the use of correlated features. If β is set to 0, the algorithm selects individual features that have maximal mutual information with the class labels, without considering the redundancy among the features. Essentially, the above process selects the feature that adds most new information about the class given the features that have been chosen already.

4.4.2 Classification-category prediction

We analyze the effectiveness of video classification performance and optimal adaptation prediction in this section. We apply supervised machine learning techniques to develop classifiers. The training data consist of samples of video clips, each represented by the features extracted from the clip. The learned classifier takes the extracted features for each new video as input, and then predicts the class that the input video most likely belongs to. Once the class is predicted, adaptation preference information of each class (as shown in Figure 4.14) is used to predict the optimal adaptation operation under different bandwidth conditions.

In our experiment, each observation was extracted from a subclip with one-GOP length, resulting in a total of 2275 observations. Each subclip carried the same category label as that of the source video clip. The results reported below were based on the average over 10 runs. In each run, a cross-validation scheme was employed, where 80% of the observations were randomly chosen as the training pool and the remaining the testing pool. We adopted the Support Vector Machine (SVM) as our classification technique. We set the SVM parameters RBF kernel $\gamma = 2$ and non-separable penalty $C = 100$.

Figure 4.15 shows the performance of the classification based on the video categories defined using r_{MAB} (the partition based on the unsupervised clustering gave similar results). The MIFS with different β values are compared with the results using the full set (i.e., all the 86 features) and a subset randomly chosen from the 86 features. Several interesting observations are found. Firstly, the full set of 86 features yields satisfactory accuracy (above 90%). As a comparison (results not listed here), using the set of about 1200 raw features resulted in accuracy below 70%. Secondly, it is clear to see the impact of β on the performance. When the number of selected features is small, MIFS works better than random and the results using non-zero β values are slightly better than that with $\beta = 0$. When more features are selected, the superiority of MIFS over random degrades. Moreover, except for $\beta = 0$, random feature selection beats MIFS after certain point (e.g., for $\beta = 0.5$ after about 10 features). This is because β penalizes the case when selected features have high correlations. It remains to be an interesting research issue how to automatically select a suitable β value. Lastly, by selecting up to 20 features through MIFS, the performances reach a saturation platform (~88%). By using alternative feature selection algorithm [44] we also reached similar conclusions. A reduced set of features will improve the efficiency of the algorithm.

Figure 4.16 further summarizes the classification performance using different sets of features. Six results are listed: the full set of 86 features, 20 features selected by MIFS with $\beta = 0$, 20 random features (randomly selected in each run from the whole set of 86 features), 20 block size features, 36 motion magnitude features, and 30 and residual energy features. This comparison helps us better understand the behavior of different content characteristic. Each component has its own contribution - block size and residual energy reflect both motion and texture information and therefore seem to add additional power compared with the motion magnitude features. In other words, the weak performance by using motion magnitude indicates the importance of incorporating the spatial (texture) information in the prediction

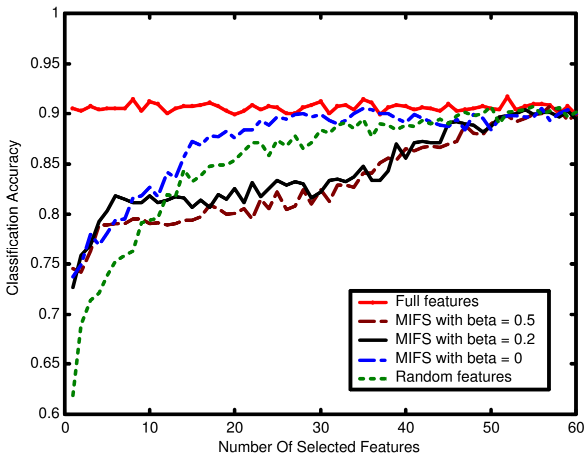


Figure 4.15: Classification Performance.

procedure. Among the reduced feature sets, the 20 features by MIFS work best, including 10 for the block size, 2 for motion magnitude, and 8 for residual energies.

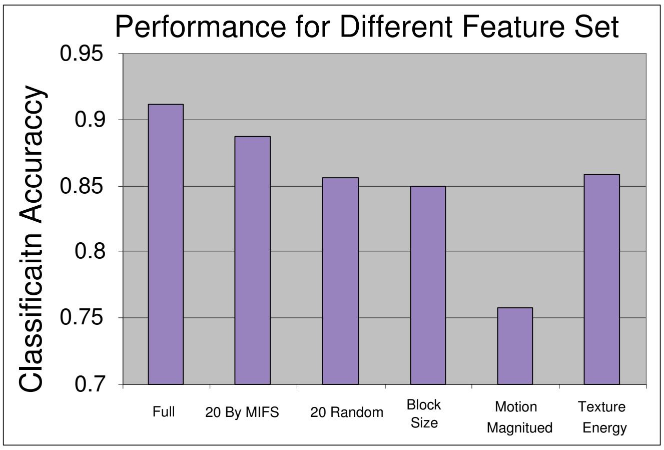


Figure 4.16: Performance comparison among different feature sets.

4.4.3 Classification - prediction of the optimal adaptation operation

The category classification is not our ultimate goal. Our aim is to use the prediction result to guide selection of the optimal MDA operation. Within each category, we can obtain the category-specific preference histogram (see Figure 4.14). Given the histogram, a straightforward prediction method is to choose the operation that is ranked best most frequently based on the preference scores given by human subjects. We measure the operation prediction accuracy (OPA) in the following way.

$$OPA = \frac{\text{Number of correct prediction}}{\text{Total number of observations}} \tag{4.6}$$

A prediction is considered correct when the predicted operation matches the actual preferred operation or one of the preferred operations in a tie. Figure 4.17 is the result of OPA over different bandwidths. As a comparison, four different approaches are analyzed: *MAB Clustering/Classification* used r_{MAB} in both clustering and classification routine (i.e., a pure domain knowledge based approach); *MAB Clustering/Content-based Classification* used categories defined through r_{MAB} and applied content features for classification; *Unsupervised Clustering/Content-based Classification* used unsupervised clustering for class definition and content-based classification; and *No Classification* predicted the operations using the preference histogram over the entire video pool. A notable improvement gain (up to 30%) can be observed by applying classification-based prediction, especially at the low bandwidth end where practical UMA applications focus most. Among different classification approaches, *MAB Clustering/Classification* outperformed other methods. This is reasonable as the MAB categories are defined based on domain-specific knowledge and each new video clip can be classified using the r_{MAB} value without errors. In comparison, two content-based classification methods performed almost as well as the MAB classification method at the high-bandwidth region ($200Kbps$ and above), but not at the low-bandwidth region (50 and $100Kbps$). Between the two content-based methods, the performances were quite close at different bandwidths, except $100Kbps$ where the approach using domain-specific knowledge was better. It is also clear that for different bandwidths, the prediction performance varies, reaching the lowest in the medium bandwidth range (200 $600Kbps$). This phenomenon comes from the fact that at mid bandwidths human subjects do not show consistent preferences to specific dimensions among different spatio-temporal scales, resulting in a large variance in the subjective scores. Figure 4.18 shows the entropy estimation of the preferred operation rankings over different bandwidths. Such estimates in some degree can be considered as the measurement of the prediction difficulty. Actually, the performance of our proposed method matches the entropy measure very well, while the approach without classification cannot. To some extent, this also validates the approach of the content-based prediction.

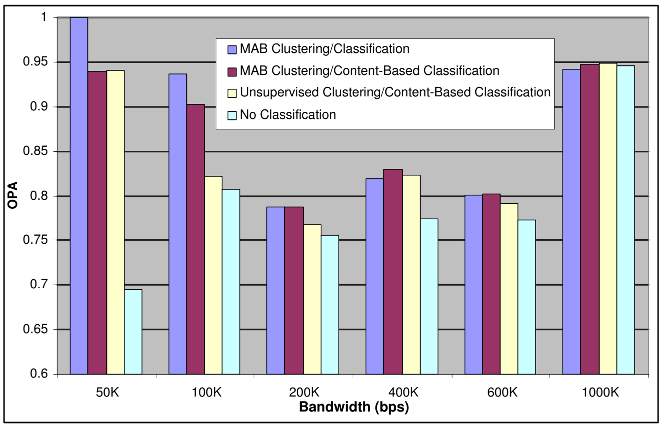


Figure 4.17: Operation prediction accuracy.

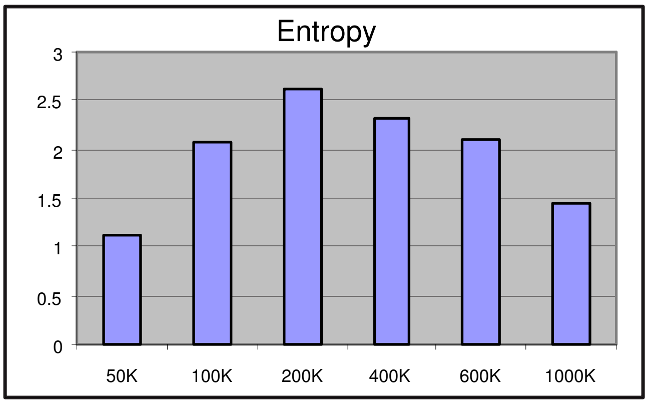


Figure 4.18: Entropy of the distributions of preferred operation at different bandwidths.

4.4.4 Computational complexity analysis

Computational complexity of the proposed system is very important for a real time application scenario. Because the MC-EZBC coder we used was not a real-time implementation, we were not able to provide the real time benchmark data. However, all of the computation processes in our system can be easily verified to be lightweight. As shown in Figure 2.11 the main costs in our system include feature extraction and classification. The classification process is very efficient. For example, SVM classification only needs to calculate the kernel function and dot product between the content features and a sparse set of support vectors. For feature extraction, r_{MAB} can be easily retrieved by parsing the encoded bit stream. In the methods using low-level content features, partial bit stream decoding is needed to obtain motion vectors, block size and wavelet coefficients, plus some minor extra calculation such as histogram counting (for motion magnitude and block size) and subband energy calculation (for residual energy). The combination of all these computations is still much lighter than the complexity of a regular decoder (because a much more complex process, motion compensation, is not needed). Considering video decoders can be implemented on most platforms with a real-time performance, it is reasonable to claim that our system can be implemented in a real-time fashion. In some special cases, the content features can even be computed at the encoder and transmitted to the adaptation decision module as side information, thus at no additional cost.

4.5 Summary

In this chapter, we address the issue of optimal MDA operation selection for scalable video codec MC-EZBC. The characteristics of SNR-spatio-temporal scalability in MC-EZBC is analyzed. In order to formulate utility function, we conducted large-scale subjective studies to evaluate the perceptual quality of videos adapted at different SNR-temporal scales. Rigorous methods were applied to assess the statistical significance of the experimental data. The generated subjective UF can be adopted to select the optimal MDA operation efficiently. The content-based prediction framework is also applied to MC-EZBC. The experiment results indicate that our proposed method can effectively reveal the relationship between the content characteristic and the MDA behavior, and therefore accurately predict the optimal adaptation operation with accuracy from 77% to 95% over different bandwidth. To the best of our knowledge, this is the first work investigating the relations between the optimal spatio-temporal adaptation operation and the content characteristics, using the subjective quality evaluation metric. It also represents the original work exploring the optimal adaptation over multi-dimensional scalable video codec like MC-EZBC.

Chapter 5

Complexity Adaptive Motion Estimation and Mode Decision for H.264 Video

5.1 Introduction

Most of today's video coding systems encode the video bit streams to achieve the best video quality (e.g., the minimal signal distortion) while satisfying certain bitrate constraints. We repeat below the optimization problem formulation in Equation (1.1):

$$\begin{aligned} &\min_{\mathbf{P}} D(\mathbf{P}) \\ &s.t., R(\mathbf{P}) \leqslant R_T \end{aligned} \tag{5.1}$$

The solution of the above problem aims at finding the optimal control parameters (CP) $\mathbf{P}$, subject to the constraints on R. Equation (5.1) does not explicitly model the required complexity in video encoding or decoding. As a matter of fact, many recent advances in the coding efficiency are accomplished by using increasingly complex computational modules, such as sophisticated processes for motion estimation. Specifically, the emerging video coding standard H.264 achieves significant advances in improving video quality and reducing bandwidth compared, but at the cost of greatly increased computational complexity.

On the contrary, many media application devices such as mobile handheld devices are getting smaller and lighter. The computational resources available on the handheld devices become relatively scarce, given the increasing functionalities and complexity of applications running on the devices. Therefore, recently in the literature there is growing interest in complexity (power) aware video coding solutions. ARMS and National Semiconductor develop a systematic approach called PowerWise technology, which can efficiently reduce the power consumption of mobile multimedia applications through adaptive voltage scaling (AVS) [65]. Chen *et al* proposed a VLSI architecture to support fast motion estimation for H.264 in [10]. In H.264, since motion estimation (ME) dominates the encoding cost (up to 90% [10]), lots of work tried to cut down the ME cost from many aspects. Tourapis extended his EPZS (Enhanced Predictive Zonal Search) algorithm to reduced the ME cost in H.264 [85]. Yin *et al* proposed a fast mode decision method to speed up ME procedure [104]. Su and Sun used a fast multiple reference frame method taking account of the correlation of motion vectors among different reference frames [69]. Zhou *et al* and Lee *et al* seperately implemented H.264 decoders based on Intel's MMX technique and single-instruction-multiple-data (SIMD) architecture to reduce the decoding complexity and improve the H.264 decoding speed by up to three times [40, 111]. In [60] Ray and Radha proposed a method to reduce the decoding complexity by selectively replacing the I-B-P Group of Pictures (GOP) structure with one using I-P only. Lengwehasatit and Ortega developed a method to reduce the decoding complexity by optimizing the Inverse DCT implementation [42]. He *etal* optimized the power-rate-distortion performance by constraining the sum of absolute difference (SAD) operations during the motion estimation process at the encoder [29].

In this chapter we focus on an important aspect of the complexity minimization problem in H.264 - how to develop an encoding algorithm that achieves both high video quality and low decoding complexity while satisfying the bit rate constraint. The goal of this work is to reduce the complexity requirement in H.264 decoding, which is quite different from other work in the literature. Specifically, we modify the video encoding algorithm to minimize the required complexity at the decoder, not the encoder. We are interested in decoding complexity deduction because we believe in the near future the mobile/wireless devices are more likely to be video content consumers instead of creators considering their limitations in power supply and computation capability. Our approach does not require any change in the existing decoder implementations. Instead, we modify the non-normative parts of the H.264 encoding algorithm to generate bit streams that can be decoded by standard-compliant decoders. In other words, we develop novel H.264 encoding algorithms that generate low-decoding-complexity and high-quality bit streams. Other techniques for the decoder power minimization, such as those in [21, 40, 42, 60, 111], are complementary and can be combined with our solution.

Specifically, when considering the decoder's complexity during video encoding, we reformulate the optimization problem as follows.

$$\begin{aligned} &\min_{\mathbf{P}} D(\mathbf{P}) \\ &s.t., R(\mathbf{P}) \leqslant R_T \\ &s.t., C(\mathbf{P}) \leqslant C_T \end{aligned} \tag{5.2}$$

where C is the computational complexity at the decoder. Compared with the problem defined in Equation (5.1), a constraint on computational complexity is explicitly added. The solution for Equation (5.2) needs to determine the best control variables, **P**, for each coding unit. Similar to the case for Equation (5.1), the control variables include quantization parameter, block mode of the motion compensation process, and the associated motion vectors.

Among the control variables, the motion vectors have the largest impact on the decoding complexity. Motion vectors can be of integer or fractional values corresponding to a displacement distance of integral pixels or fractional pixels. When a motion vector is of a sub-pixel value, multi-tap filtering is required to compute interpolation to form a reference block that is needed in the motion compensation process in the decoder. Such interpolation filtering involves huge computational cost and significantly increases the overall decoding complexity. Figure 5.1 shows the breakdown of the complexity of a typical H.264 decoder implementation [38]. It is clear that the interpolation component constitutes about 50% of the decoding complexity. Although for mobile multimedia applications there are other power consuming components like wireless communication, display, and memory access, the decoding process is typically a significant one. Therefore improving the cost associated with the interpolation process is important for achieving a low-power decoding system, either in hardware or software.

In this work, we extend the conventional rate-distortion framework based on the Lagrange optimization method to incorporate the computational complexity. To estimate the complexity associated with different types of motion vectors, we develop models to approximate the implementation cost involved in the interpolation filtering process. In addition, we extend the rate control algorithm to handle the joint rate-complexity control issue so that both the targets of rate and complexity can be met. Our optimization method intelligently selects the block mode and motion vector type of each coding unit to achieve the highest video quality. When tested

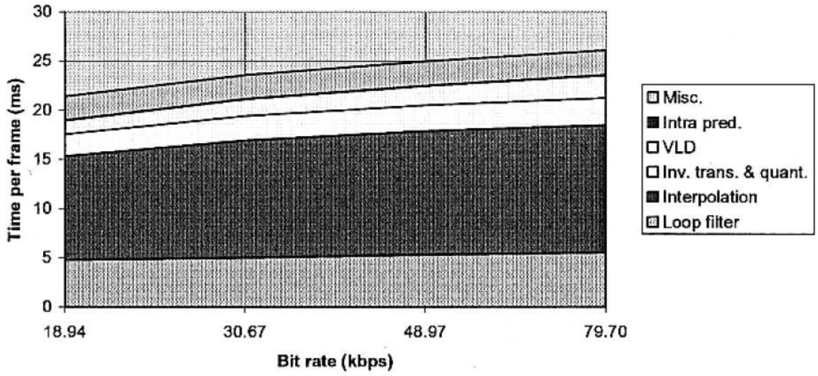


Figure 5.1: Breakdown of computational complexity distribution in a typical H.264 decoding process. The figure is based on decoding of the *Foreman* test video sequence with the QCIF resolution.

over a diverse set of video sequences over different bit rates, our solution achieves very significant complexity reduction (up to 60%) of the most complex component, interpolation filtering, while keeping the video quality almost intact (degradation within 0.2dB). When incorporated into the practical system, our solution has great potential in reducing the overall power consumption.

The rest of this chapter is organized as follows. Section 5.2 includes reviews of principle components of a typical hybrid video coding system based on which H.264 is built. It describes the basic concepts of motion estimation, motion compensation, and their implication on the computational complexity. The rate-distortion optimization framework based on the Lagrange optimization method is reviewed. It also explains the process used to control the rate over frames to meet the overall target. In Section 5.3, we present the proposed CAMED method for generating low-complexity bit streams. Section 5.4 includes the experiment results. Summarization is provided in Section 5.5.

5.2 Overview of H.264

In this section we provide an overview of H.264 coding system. We only address the parts related with the scope of this chapter. A thorough introduction on H.264 can be found from [71, 84].

5.2.1 Review of typical hybrid video coding systems

Figure 5.2 illustrates the system diagram for a typical hybrid motion compensation and block-transform video coding system. The darker box shows the decoding procedure, which is also simulated in the encoder system for rate control purpose. The basic decoding unit is a macroblock (MB). For each MB, the encoded bit stream first undergoes entropy decoding to obtain the syntax bits (not shown in the figure), motion vector $\mathbf{V}$, and quantized coefficients $\bar{d}_T(t)$, where t is the time index of the image frame. Typical entropy codecs include variable length coding (VLC) and adaptive arithmetical coding (AAC). Inverse quantization is then employed to obtain the transform coefficient $d_T(t)$, which is further fed to an inverse transform module to reconstruct the pixel value or prediction error d_t, depending on whether intro- or inter-coded mode is utilized during encoding. For inter-coding mode, motion compensation is applied to generate the reference image $\bar{P}_R(t)$ using motion vector $\mathbf{V}$ and previously decoded and buffered reference image $P_R(t-1)$. We use motion compensation to refer to the process of compensating the image displacement due to motion across frames. When the motion vector is of a sub-pixel value, interpolation is needed to compute the reference image. Lastly, by combining the prediction error $d(t)$ and the reference image $\bar{P}_R(t)$ the decoded image of the current frame is output.

The computational complexity of each component varies. Some are relatively constant and independent of the encoded data while others heavily depend on the

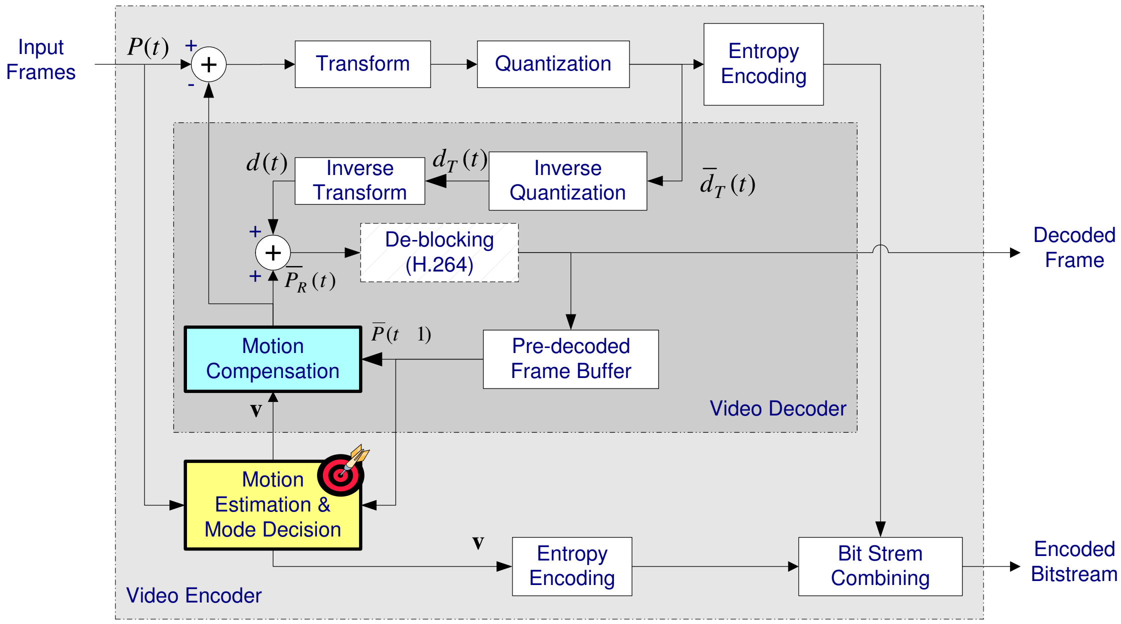


Figure 5.2: Conceptual diagram for typical video coding systems.

coding results. For example, inverse quantization has nearly fixed computational cost per coding unit while the motion compensation component has variable complexity depending on the block mode and the type of motion vector. Furthermore, as shown in Figure 5.1, the decoder complexity is dominated by the interpolation filtering process used in motion compensation if the motion vectors are sub-pixel. Other parts of the decoding system, like entropy decoding and inverse transform, do not incur significant computational cost when compared to the interpolation process.

Note motion estimation is usually the most computationally complex process since it involves searching over a large range of possible reference locations, each of which may require interpolation filtering. Recognizing this, many fast motion estimation algorithms such as those proposed in [11, 13, 69, 85, 103, 104] have been developed to reduce the motion estimation complexity during encoding. Other work proposes scalable methods for motion estimation [64] to control the coding complexity. Nevertheless these methods all focused on the encoding complexity reduction instead of the decoding complexity.

5.2.2 Sub-pixel interpolation

Motion estimation is one of the most important components, and also the most computationally complex part in any video coding systems. Motion estimation can be illustrated using Figure 5.3. The basic idea is to search for an optimal block with similar values in previous coded frames as the reference signal for the block in current frame so that the encoding cost can be minimized. The optimal reference signal position is indicated by the displacement vector, called motion vector (denoted as V in Figure 5.3). Motion estimation applies the basic idea of inter-frame predictive coding. Sometimes, multiple reference signals are used to form motion estimation, like the case for bi-directional inter-frame prediction. Motion vectors are entropy encoded in a differential and predictive manner [100]. Compared to motion estimation, motion compensation is the procedure by which the decoder extracts a reference signal from the location indicated by the motion vector. In reconstructing the reference signal, interpolation is a widely adopted technique to improve the compensation precision when the motion vector has a sub-pixel value. The effectiveness of the sub-pixel motion compensation has been verified in H.263 and subsequent coding standards, at the cost of increasing complexity (up to 50% referring to Figure 5.1). Therefore reducing the motion compensation complexity becomes the most important target for improvement.

H.264 uses up to quarter pixel precision during interpolation [98, 100]. Figure 5.4 illustrates the details of this procedure, where gray blocks with capital letters indicate the integer locations and the white blocks with lowercase letters the sub

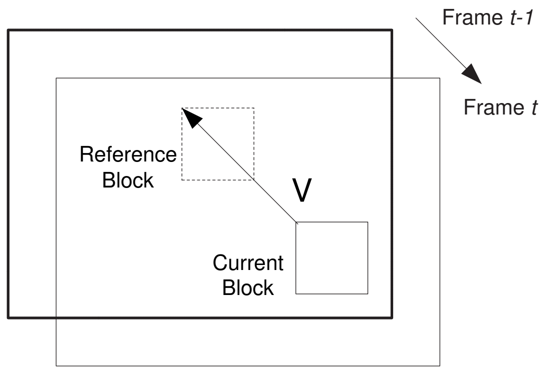


Figure 5.3: Motion compensation between current and the reference frames.

pixels. All half-pixel locations undergo 6-tap FIR filtering horizontally and vertically, whenever any one applies. All quarter-pixel locations undergo 2-tap average filtering using integer and half pixels. For example, the following formulae are used to calculate sub pixel b and e:

$$
\begin{aligned}
b &= ((E - 5F + 20G + 20H - 5I - J) + 16)/32 \\
e &= (b + h + 1)/2
\end{aligned}
$$

The amount of filtering operations varies depending on the exact location of the pixel. Table 5.1 lists the possible interpolation operations and the associated complexity. It is clear that different interpolation methods have quite different computing complexities.

Given the information about the interpolation cost associated with each type of motion vectors, the basic idea of reducing the decoder complexity is to select motion vectors that involve less interpolation complexity while keeping the video quality high. Our empirical analysis of some H.264 statistical data shows that depending on the video content, 40% to 80% of motion vectors are located on sub pixels with different interpolation complexities. Therefore the principal approach to complexity reduction is to change motion vectors from high complexity sub pixel positions into the ones with low complexity, or even to integer-pixel positions.

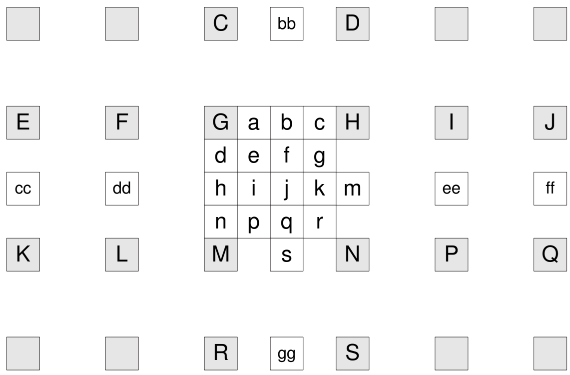


Figure 5.4: Notations for sub-pixel locations in H.264.

Table 5.1: Sub pixel locations and their interpolation complexities

Sub Pixel Type	Points	Interpolation
(0, 0)	G	No
(0, 1/2), (1/2, 0)	b, h	1 6-tap
(0, 1/4), (1/4, 0), (0,3/4), (3/4, 0)	a, c, d, n	1 6-tap + 1 2-tap
(1/4, 1/4), (1/4, 3/4), (3/4, 1/4), (3/4, 3/4)	e, g, p, r	2 6-tap + 1 2-tap
(1/2), (1/2)	j	7 6-tap
(1/2, 1/4), (1/4, 1/2), (3/4, 1/2), (1/2, 3/4)	i, f, k, q	7 6-tap + 1 2-tap

5.2.3 Block mode

In order to further reduce the temporal redundancy and improve the efficiency of motion estimation, H.264 defines a diverse set of block mode options. Besides the conventional modes of intra, forward, backward, bi-directional, two new important modes are introduced: variable block size and SKIP/DIRECT.

Firstly, unlike earlier coding standards using a fixed block size (usually 16x16 or 8x8) during motion estimation, H.264 allows to partition an MB into into several blocks with variable block size, ranging from 16 pixels to 4 pixels in each dimension. The possible modes of different block sizes are shown in Figure 5.5. An MB can comprise up to 16 blocks. Each block with reduced size can have its individual motion vectors to estimate the local motion at a finer granularity. Though such finer block sizes incur overhead such as extra computation for searching and extra bits for coding the motion vectors, they allow more accurate prediction in the motion compensation process and consequently the residual errors can be considerably reduced, which are usually favorable for the final rate-distortion performance.

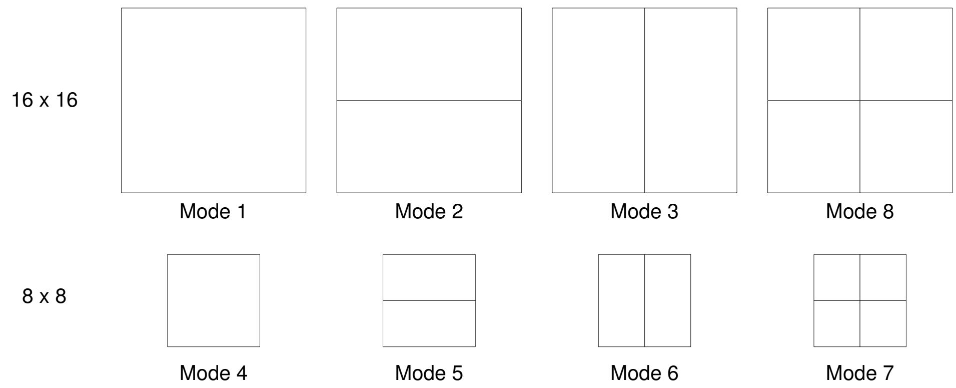
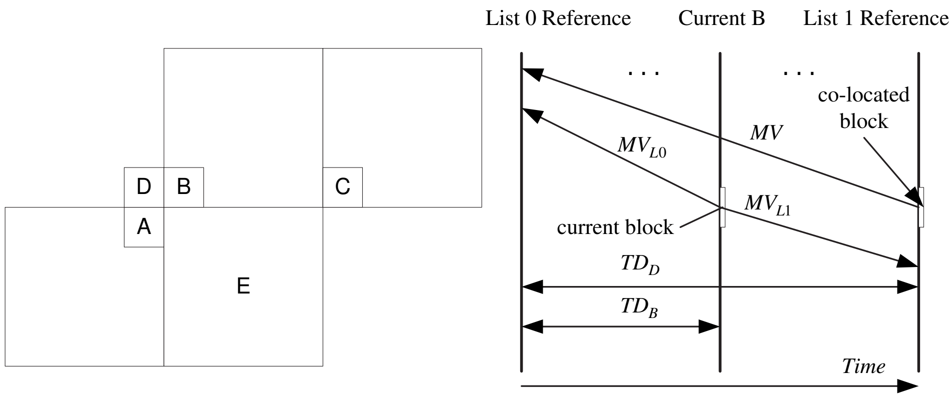


Figure 5.5: Modes of variable block sizes in H.264.

Secondly, the SKIP/DIRECT mode is utilized for the P/B frame in H.264 motion compensation to further increase the coding efficiency. The basic idea is to use the spatial/temporal neighbor motion vectors to predict the motion vector of the current block, without sending extra bits to encode the current motion vector. Figure 5.6(a) illustrates the SKIP mode, where the motion vectors of blocks A, B, C and D (if available) may be used to estimate the motion vector of MB E. In Figure 5.6(b) the motion vector of the current block in a B frame is interpolated from the motion vector of the co-located block from the adjacent frames, assuming a constant global motion. Details regarding the SKIP/DIRECT mode can be found in [86, 100].

In our mode decision algorithm to be described later, both the variable-size block mode and the SKIP/DIRECT mode are considered during the search process.



(a) Skip Mode (b) Direct Mode

Figure 5.6: The SKIP/DIRECT mode for the P/B frame in H.264.

The selection of block mode has direct impact on the decoder computational complexity, because it determines what kind of motion vectors is recorded in the bit stream. Optimal selection of the block mode and the associated motion vectors is the main problem addressed in our work, where a systematic solution is derived.

5.2.4 Motion vector searching and block mode selection

As introduced in Section 5.1, conventional video coding systems encode the video bit stream by solving the optimization problem defined in Equation (5.1). The main control variables **P** involved in this procedure include motion vector V, block mode M and quantization parameter QP. There is complex interaction between the choices of these variables and thus finding the optimal solution is a typical non-polynomial (NP-hard) problem. In practice, compromised approaches are taken and approximate solutions are developed. For example, typically QP is determined through some empirical models and updated throughout the video sequence by some rate control algorithms. Given QP, other variables, such as motion vector and block mode, are decided by applying some rate-distortion optimization process. An excellent survey of these procedures is described in [70]. We present a brief summary in the following.

Specifically, for each block B with a block mode M, the motion vector associated with the block is selected through a rate-distortion joint cost function [70]:

$$\begin{aligned} \mathbf{V}^*(B,M) &= \arg\min_{\mathbf{V}} J^{R,D}_{MOTION}(\mathbf{V}|B,M), \mathbf{V} \in \sup\{\mathbf{V}\} \\ &= \arg\min_{\mathbf{V}}\{D_{DFD}(\mathbf{V}|B,M) + \lambda_{MOTION} R_{MOTION}(\mathbf{V}|B,M)\} \end{aligned} \tag{5.3}$$

where $\mathbf{V}^*$ is the optimal motion vector, $\sup\{\mathbf{V}\}$ defines the search space, whose dimensions include the prediction direction, the reference frame list and the search range. R_{MOTION} is the estimated bit rate to record the motion vector. D_{DFD} represents the prediction error between the current block and the reference block. Usually the sum of absolute difference (SAD) is adopted because the search space of motion vector is much larger than that of mode and SAD has lighter computation cost compared with the sum of squared difference (SSD). $J^{R,D}_{MOTION}(\mathbf{V}|B,M)$ is the rate-distortion joint cost comprising of R_{MOTION} and D_{DFD}. λ_{MOTION} is the Lagrange multiplier to control the weight of the bit rate cost, relative to the signal distortion caused by the prediction error.

In a similar manner the block mode for an MB is decided by the following.

$$\begin{aligned} M^*(MB,QP) &= \arg\min_{M} J^{R,D}_{MODE}(M|MB,QP), M \in \sup\{M\} \\ &= \arg\min_{M}\{D_{REC}(M|MB,QP) + \lambda_{MODE} R_{REC}(M|MB,QP)\} \end{aligned} \tag{5.4}$$

where M^* is the optimal block mode, and $\sup\{M\}$ is the set of block mode options (such as INTRA, SKIP, DIRECT, FORWARD, BACKWARD, BIDIRECTION, etc). A full list of block mode options in H.264 can be found in [86]. D_{REC} is the SSD between the current MB and the reconstructed one through motion compensation. R_{REC} is the estimated bit rate associated with mode M. $J^{R,D}_{MODE}(M|MB, QP)$ is the joint cost comprising of rate R_{REC} and distortion D_{REC}, and λ_{MODE} is the Lagrange multiplier. The motion vectors associated with the optimal block mode $\mathbf{V}^*(B, M^*)$ will be the final coded data recorded in the bit stream.

The Lagrange multipliers used in the above two cost functions determine the relative weights between signal quality and bit rate. To simply the search process, an empirically derived relationship as the following is typically used in practice. The square root relationship is partly due to the fact that SAD is used in modeling D_{DFD} while SSD is used for D_{REC}.

$$\lambda_{MOTION} = \sqrt{\lambda_{MODE}} \tag{5.5}$$

5.2.5 Rate control

Rate control (RC) is the procedure of adjusting control parameters (mainly QP) so that the target rate requirement can be achieved while optimizing the overall video quality. Given a target bit rate, we can compute the average allocated bit rate for each basic coding unit. Then we can use the Lagrange optimization method to find the optimal set of control variables. However, searching over the entire variable space is infeasibly complex. In practice, most implementations use empirical models to restrict the search space. For example, a popular method, called rate-quantization modelling, maps the target bit rate to the quantization parameter, from which the Lagrange multipliers are decided. In addition, since coding of a data unit may not result in a bit rate that exactly matches the target, a separate process, called buffer management, is used to monitor the available bit rate budget for the remaining data units and thus update the allocated recourse. We briefly review these processes in the following.

Rate-Quantization (R-Q) model describes the relationship between and the bit rate. A widely adopted quadratic R-Q model is [14]:

$$R = D(P_1 \cdot QP^{-1} + P_2 \cdot QP^{-2}) \tag{5.6}$$

where D is the source complexity of the video signal, and usually measured using the motion estimation prediction errors (such as SAD), and $\{P_1, P_2\}$ are model parameters. Some systems use $P_2 = 0$ for simplicity. A typical R-Q modeling procedure involves two major steps: model estimation and QP prediction. Figure 5.7 shows a conceptual illustration of these procedures. Firstly several basic coding units are coded using some preset QP values. The coding units may include a certain number of MBs or one whole frame. The resulting rate-quantization-distortion (R-Q-D) points are collected, as indicated by the gray circles in Figure 5.7. The model in Equation (5.6) is then estimated based on the observations. The estimated model is indicated by the multiple curves shown in Figure 5.7. The estimated model can then be used to determine the QP value for the next coding unit based on the target bit rate R_t and source complexity D_t for the new unit. The former is decided by the buffer management process to be described below, and the latter is predicted using previous observations of the source complexity. Usually the source complexity is assumed to vary gradually and can be estimated using some simple relationship (such as linear predictive coding, or LPC). Once coding of the new unit is completed, new observations of the R-Q-D points are collected and used to update the estimation of the RQ model in a sliding window manner. Namely, the oldest R-Q-D point is purged and the latest point is added to update the model.

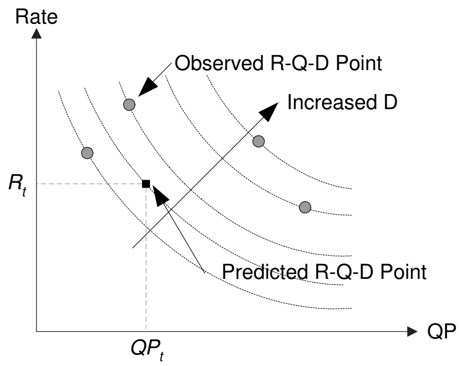


Figure 5.7: Rate-Quantization model estimation and QP prediction in the rate control process.

The buffer management employs a virtual buffer to simulate the behavior of the data buffer on the decoder side. It is an important component in rate control in order to adjust the target bit rate for each coding unit and avoid the problem of buffer overflow or underflow. For example, given a target bit rate for the video sequence, the average bit rate allocation for each Group of Pictures (GOP) can be computed, and the allocated bit rate for a new frame to be coded (such as P frame) can be determined by monitoring the actual number of bits spent on the previous frames.

In H.264, given the target rate and QP for the coding unit, the following empirical relationship is used to determine the Lagrange multiplier needed in the rate-distortion tradeoff optimization.

$$\lambda_{MODE} = 0.85 \times 2^{\frac{QP-12}{3}} \tag{5.7}$$

The validity of such a model is justified by empirical simulations, though some analytical explanations have been offered in the literature such as [70]. Such an empirical model is very useful in simplify the search process in the Lagrange optimization method, while practical implementations have often shown satisfactory performance. Other parameters like can also be found according to Equation (5.5).

5.3 Complexity Adaptive Motion Estimation and Mode Decision

We propose a new system for Complexity-Adaptive Motion Estimation and mode Decision (CAMED). Given defined metrics for signal distortion and computational complexity, CAMED method explores the tradeoff between video quality and resource consumption (both bit rate and computational complexity) to approximate the optimal motion vectors and block mode used in the motion compensation process in the decoder. CAMED system consists of several components: the rate-distortion-complexity (R-D-C) joint optimization framework, the complexity cost function, and the complexity control algorithm. The R-D-C framework extends the previously discussed Lagrange optimization framework to incorporate the complexity term. The complexity cost function provides quantitative measurements of the required computation for each motion vector type. The complexity control algorithm is used to control the complexity over different coding units to meet the overall target complexity. We discuss each of them in the following.

5.3.1 The Rate-Distortion-Complexity optimization framework

Our proposed CAMED system basically tries to solve the problem defined in Equation (5.2), with an explicit Lagrange term to model the complexity cost. Therefore, the motion vectors are selected through a rate-distortion-complexity joint cost function:

$$\begin{aligned} \mathbf{V}^*(B,M) &= \arg\min_{\mathbf{V}} J^{R,D,C}_{MOTION}(\mathbf{V}|B,M), \mathbf{V} \in \sup\{\mathbf{V}\} \\ &= \arg\min_{\mathbf{V}} \{J^{R,D}_{MOTION}(\mathbf{V}|B,M) + \gamma_{MOTION} C_{MOTION}(\mathbf{V}|B,M)\} \end{aligned} \tag{5.8}$$

where C_{MOTION} is the complexity cost function associated with the selected motion vector $(\mathbf{V}|B,M)$, γ_{MOTION} is the Lagrange multiplier for the complexity term, $J^{R,D}_{MOTION}$ is the rate-distortion joint cost function defined in Equation (5.3), and $J^{R,D,C}_{MOTION}$ is the rate-distortion-complexity joint cost function.

Similar to the earlier case described Equation (5.4), the block mode search process is guided by the following.

$$\begin{aligned} M^*(MB,QP) &= \arg\min_{M} J^{R,D,C}_{MODE}(M|MB,QP), M \in \sup\{M\} \\ &= \arg\min_{M} \{J^{R,D}_{MODE}(M|MB,QP) + \gamma_{MODE} C_{MODE}(M|MB,QP)\} \end{aligned} \tag{5.9}$$

where C_{MODE} is the complexity cost function associated with the block mode, γ_{MODE} is the Lagrange multiplier, $J^{R,D}_{MODE}$ is the rate-distortion joint cost function defined in Equation (5.4), and $J^{R,D,C}_{MODE}$ is the rate-distortion-complexity joint cost function.

Now consider two extreme cases. When $\gamma_{MODE} = 0$, the solutions of Equation (5.8) and Equation (5.9) are identical with the ones in Equation (5.3) and Equation (5.4), namely no consideration was given to the complexity constraint and many motion vectors may be of sub-pixel values in order to minimize the distortion. When $\gamma_{MODE} = \infty$, all motion vectors are forced to integer pixel locations in order to minimize the complexity involved in interpolation for sub-pixel locations. Clearly there is a tradeoff between these two extremes to balance the performance in terms of quality and complexity.

For simplification, we adopt restrictions like that described in Equation (5.5) to limit the search space. In our experiments to be described later, we use the following relationship to link γ_{MODE} and γ_{MOTION}.

$$\gamma_{MOTION} = \sqrt{\gamma_{MODE}} \tag{5.10}$$

5.3.2 Complexity cost function

5.3.2.1 Platform independent modelling

In the joint cost function described above, we need a quantitative model to estimate the complexity associated with each candidate motion vector and block mode. As discussed in Section 5.2.2, the computational complexity is heavily influenced by the type of the motion vector (integer, half-pixel, or quarter-pixel) and the interpolation filters used in the motion compensation process. If we just focus on the interpolation filtering cost, quantitative estimates of such complexities can be approximated by the number of filtering operations needed in interpolation, such as those listed in Table 5.1. For example, using the same 6-tap filter and 2-tap filter implementations, the complexity of each motion vector type is as follows.

$$c_B(\mathbf{V}) = N_B \cdot c_P(\mathbf{V}) \tag{5.11}$$

$$c_P(\mathbf{V}) = \begin{cases} 0 & \mathbf{V} \text{ is integer MV} \\ e_6 & \mathbf{V} \text{ is subpixel b, h} \\ e_6 + e_2 & \mathbf{V} \text{ is subpixel a, c, d, n} \\ 2e_6 + e_2 & \mathbf{V} \text{ is subpixel e, g, p, r} \\ 7e_6 & \mathbf{V} \text{ is subpixel j} \\ 7e_6 + e_2 & \mathbf{V} \text{ is subpixel i, f, k, q} \end{cases} \tag{5.12}$$

where $c_B(\mathbf{V})$ is the computational cost for the current coding block, $\mathbf{V}$ is the motion vector, $c_P(\mathbf{V})$ is the computational complexity required for calculating a reference pixel, N_B is the number of pixels in the current coding block, and e_6, e_2 are the estimated complexities for 6-tap and 2-tap interpolation respectively. Our experiment later will show that a simplified model ignoring the 2-tap interpolation will mostly result in the same selection of the motion vectors. With such simplification, the above model becomes the following with a common factor e_6 removed.

$$c_P(\mathbf{V}) = \begin{cases} 0 & \mathbf{V} \text{ is integer MV} \\ 1 & \mathbf{V} \text{ is subpixel a, b, c, d, h and n} \\ 2 & \mathbf{V} \text{ is subpixel e, g, p, r} \\ 7 & \mathbf{V} \text{ is subpixel i, j, f, k, q} \end{cases} \tag{5.13}$$

Equation (5.11) and (5.13) estimate the computational complexity based on the interpolation operation following the standard description [84]. We call these platform-independent modelling. Alternatively the complexity cost can be derived from the specific software or hardware implementations, so called platform-dependent modelling. We will introduce two platform-dependent models as follows.

5.3.2.2 Block-based complexity modelling

The complexity cost functions defined in Equation (5.11) and (5.13) are also considered as pixel-based [18] in that the complexity is calculated for each pixel independently without considering the reusability of previous calculated pixel (or sub pixel) values. For block-based motion compensation as adopted in H.264, some interpolations can be saved by directly using previous computed results. Again according to the standard description (Section 8.4.2.2 in [84]), the following categories of sub pixels are considered.

1. For integer pixel no interpolation is necessary, and the complexity is zero.

2. For sub pixels a, b, c, d, h and n, they are located in either integer row or integer column, only one 6-tap filtering is necessary for them. Considering a 4×4 block (the minimum MC unit in H.264), the complexity is $1 \times 16 = 16$.

3. For sub pixels e, g, p and r, similar to previous case, the complexity is $2 \times 16 = 32$.

4. For sub pixels i, j, f, k and q, on each column within a 4×4 block, the topmost sub pixel requires full 7 6-tap interpolations. Whereas for each of the remaining three sub pixels located in the same column, 5 6-tap interpolations calculating its upper sub pixel value can be reused and only two additional 6-tap interpolations are necessary. Therefore, the complexity is $7 \times 4 + 2 \times 12 = 52$.

Therefore, block-based complexity modelling is (after value scaling):

$$c_P(\mathbf{V}) = \begin{cases} 0 & \mathbf{V} \text{ is integer MV} \\ 4 & \mathbf{V} \text{ is subpixel a, b, c, d, h and n} \\ 8 & \mathbf{V} \text{ is subpixel e, g, p, r} \\ 13 & \mathbf{V} \text{ is subpixel i, j, f, k, q} \end{cases} \tag{5.14}$$

The model in Equation (5.14) can even be further fine tuned considering variable block size implementation during MC. This results in a lookup table as shown in Table 5.2.

Table 5.2: Lookup table for complexity cost using variable block size MC implementation

Mode	Integer	1 6-tap	2 6-tap	7 6-tap
SKIP/DIRECT	0	256	512	592
16x16	0	256	512	592
16x8	0	128	256	296
8x16	0	128	256	296
8x8	0	64	128	168
8x4	0	32	64	84
4x8	0	32	64	84
4x4	0	16	32	52
Intra	0	0	0	0

5.3.2.3 Hardware-based complexity modelling

In hardware implementation, each interpolation operation can be divided into a number of basic operators such as addition, shifts, and/or multiplications. In this case, e_6, e_2 can be modelled with more details such as the following.

$$e_i = \sum_j \rho N(O_j) P(O_j), i = 2, 6 \tag{5.15}$$

where O_j is the basic operator involved in the interpolation implementation, $N(O_j)$ is the required number of operator O_j, $P(O_j)$ is the power consumption of operator O_j, and $\rho \geq 1$ is the adjustment factor to consider additional power cost such as memory access. For instance, Figure 5.8 shows a hardware implementation of interpolation that was introduced in [10]. Its estimated complexity is

$$e_6 = \rho(6P_{add} + 2P_{shift}) \tag{5.16}$$

where P_{add}, P_{shift} are the power consumption for the addition operator and the 2-bit shift operator respectively. Each block may be associated with multiple refer-

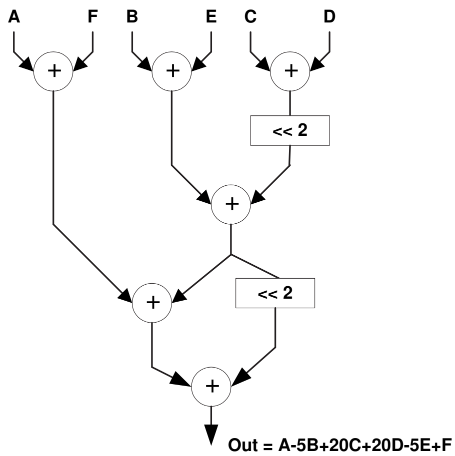


Figure 5.8: A hardware implementation for the interpolation unit.

ence blocks, each of which needs a motion vector. For example, for bi-directional prediction, each block may need two motion vectors for forward and backward prediction respectively. Thus, the computational cost for a block B with the block mode M is calculated as:

$$C_{MOTION}(\mathbf{V}|B, M) = \sum_j \Big(c_B(\mathbf{V}_j, M, B) \Big) \tag{5.17}$$

where the summation is over each reference block.

Each MB may consist of several smaller blocks, depending on the block mode, M. The overall computational cost associated with a MB and a block mode can be calculated as:

$$C_{MODE}(M|MB) = \sum_i \sum_j \Big(c_B(B_i, \mathbf{V}_j, MB) \Big) \tag{5.18}$$

where i is the index of the individual blocks contained in the MB, and j is the index for multiple motion vectors associated with a single block. Equation (5.17) and Equation (5.18) are generic and applicable to all inter-coded block modes, including foreword/backward/bi-directional motion compensation and SKIP/DIRECT. The buffer reuse mechanism described in Section 5.3.2.2 can also be considered here to fine tune the model.

In [40, 111] H.264 decoder were implemented using hardware platform acceleration such as Intel multimedia extensions (MMX) instruction and single instruction multiple data (SIMD) architecture with greatly reduced complexity. The complexity cost for these implementations need careful analysis in order to achieve correct complexity modelling. This is scheduled as future work and will not be investigated in this thesis.

5.3.3 Complexity control

Equation (5.8) and (5.9) use Lagrange multiplier to formulate R-D-C combined optimization problems. If we assume that the selection of motion vector and block mode of certain MB is independent of such behaviors in other MBs (which is a reasonable approximation of the real case), at the optimal solution each MB will have the same Lagrange multiplier $(\gamma_{MOTION}, \gamma_{MODE})$. This is an important property of Lagrange multiplier [59]. In other words, given specific $\hat{\gamma}_{MODE}$ and considering

Equation (5.10), we can obtain the bit stream with complexity $C(\hat{\gamma}_{MODE})$. This $\hat{\gamma}_{MODE}$ is (approximately) the optimal solution for the following problem:

$$\min_{\{\vec{\mathbf{V}},\vec{M}\}} \sum_{i=1}^{N} J^{R,D}(\mathbf{V_i}, \mathbf{M_i}) \tag{5.19}$$
$$s.t., \sum_{i=1}^{N} C(\mathbf{V_i}, \mathbf{M_i}) \leqslant C(\hat{\gamma}_{MODE})$$

where $\vec{\mathbf{V}} = \Big(\mathbf{V}_1, \mathbf{V}_2, ..., \mathbf{V}_N\Big)$ and $\vec{\mathbf{M}} = \Big(\mathbf{M_1}, \mathbf{M_2}, ..., \mathbf{M_N}\Big)$ are the motion vectors and block modes for all MBs respectively, $\mathbf{V_i}$ and $\mathbf{M_i}$ are the motion vector and block mode for i^{th} MB respectively, $J^{R,D}$ is the R-D cost function, and $C(\mathbf{V}, \mathbf{M})$ is the complexity cost function. Unfortunately, the complexity level $C(\hat{\gamma}_{MODE})$ associated with $\hat{\gamma}_{MODE}$ cannot be known in advance unless the bit stream has been encoded. Therefore, the Lagrange multiplier has to be adjusted in order to match certain target complexity level. We call this procedure as complexity control.

Complexity control, analogous to the rate control process described in Section 5.2.5, is a process to allocate the complexity resource among the coding units and to determine parameters like Lagrange multiplier γ_{MODE} to be used in the optimization procedure. In Section 5.2.5, the allocated bit rate is mapped to the quantization parameter, which in term is used to find the Lagrange multiplier λ_{MODE}. In this section, we describe two components of the complexity control algorithm - the complexity modeling and the buffer management. The former is used to characterize the relationship between the target complexity and the Lagrange multiplier γ_{MODE}. The latter is for monitoring the complexity usage and updating the available computational resource for each new data unit.

5.3.3.1 Complexity modelling

In complexity control a feasible modelling of complexity and control parameter (γ_{MODE} in our case) is necessary. So far there is very little knowledge regarding the statistical properties of the computational complexity in H.264. Therefore, similar to the practical solutions used in most rate control algorithms, we resort to empirical observations from experimental simulations.

One of the objectives is to find the relationship between the target complexity and the optimization control parameter, γ_{MODE}. Figure 5.9 shows some simulations results revealing such relationship. The details of the experiment will be described later in this paper. The results indicate there is an approximately linear relationship between the complexity value and log of the Lagrange multiplier. It is also clear the type of the frame (B or P) influences greatly the relationship.

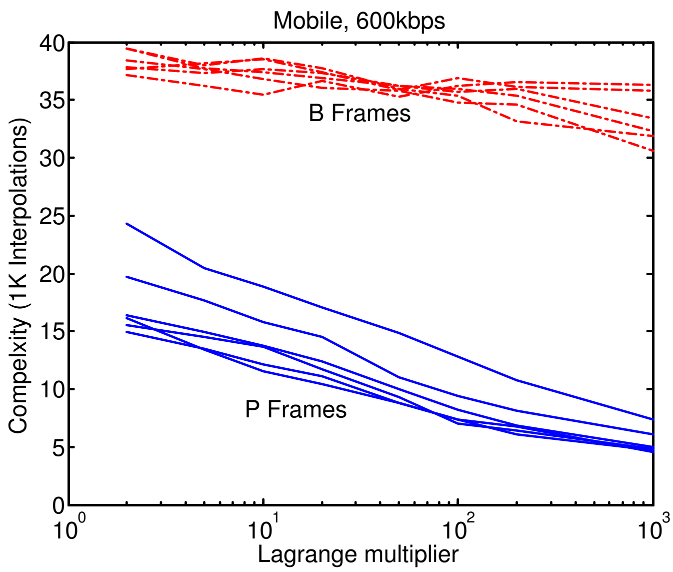


Figure 5.9: Relationship between Lagrange multiplier and the resulting complexity (top: B frames, bottom: P frames).

A reasonable model based on the above observations is as follows.

$$C_{MODE} = D\Big(K_1 \ln(\gamma_{MODE}) + K_0\Big) \tag{5.20}$$

where C is the complexity, D is a factor measuring the video source complexity similar to that used in Equation (5.6) for rate control. K_0, K_1 are the model parameters that needed to be learned during the coding procedure. Due to different coding mechanism, P and B frames will have distinguished model parameters and need to be handled separately.

Though lacking a theoretical explanation, the above model is driven by the empirical simulation observations. The linear dependence of the computational complexity on the signal source complexity is also intuitive - the more complex the signal source is, the higher accuracy is needed in estimating the motion vector and thus there is a larger gain in using sub-pixel motion vectors, resulting in an increased computational cost. Figure 5.10 shows the approximate linear relationship between the computational complexity and the mean prediction error (measured in mean absolute difference, MAD) from our simulation (details to be described in later sections). Like the previous case of rate control, the MAD measure can be considered as approximate estimation of the signal source complexity. Using the above model, the Lagrange multiplier $\gamma_{MODE}(t)$ for the current coding unit t can be determined by the following.

$$\gamma_{MODE}(t) = \exp\{\frac{C(t) - K_0 D(t)}{K_1 D(t)}\} \tag{5.21}$$

where $C(t)$ is the allocated computational budget and is the predicted complexity measurement for unit . In practice, in order to avoid large quality fluctuation, the

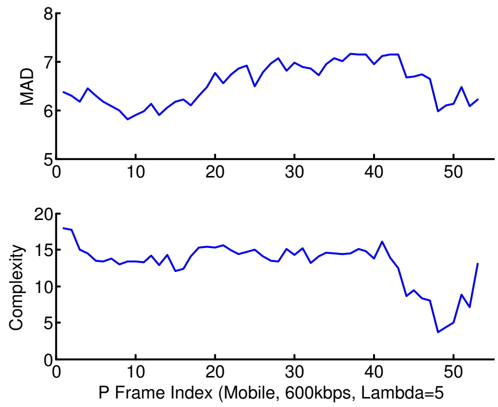


Figure 5.10: The relationship between the computational complexity and the signal source complexity.

change rate of $\gamma_{MODE}(t)$ is bounded by some thresholds.

5.3.3.2 Buffer management

Complexity buffer is a virtual buffer to simulate the complexity usage status on the decoder side. It is analogous to the rate buffer used in the rate control to update the estimation of available resource and avoid issues of buffer overflow or underflow. Denote C_{GOP} the remaining complexity budget in one GOP, N_P, N_B the remaining numbers of P, B frames respectively, and the complexity ratio between P and B, which is updated along the video coding. The target complexity levels for P, B frame C_P, C_B are calculated by solving the following equations:

$$\frac{C_B}{C_P} = \eta \tag{5.22}$$

$$N_P C_B + N_B C_B = C_{GOP} \tag{5.23}$$

Once C_P, C_B are available, $\gamma_{MODE}(t)$ is determined using the model described in the previous subsection. The formulations in Equation (5.22) and Equation (5.23) assume the basic coding unit as one frame. It can be easily extended to smaller units for a finer granularity.

5.4 Experiment Results

5.4.1 Experiment environment

Table 5.3 lists the experiment environment used in our simulation. Four standard test video sequences were chosen and they had distinguished characteristics in motion intensity and texture complexity, two crucial factors influencing the motion estimation performance. H.264 reference codec of version JM82 was used.

Table 5.3: Experiment Environment

Sequence Information	
Sequence Name	*Akiyo, Foreman, Mobile, Stefan*
Image Format	CIF (352 × 288 pixels)
Video Format	30fps, GOP15, subGOP3 (unless specified otherwise)
Simulation Parameters	
Bit Rate	100, 200, 400, 600, 800, 1000 $Kbps$
γ_{MODE} Values	From 0 to 500 (Lambda in the figure)
H.264 Configuration (selected)	
Profile	Main (unless specified otherwise)
Search Range	32
Block Mode	All on
Fast Motion Estimation	Off (unless specified otherwise)
Frame / Slice Mode	Frame
Direct Mode Type	Temporal
ate Control Unit	11 Macroblocks
S-P Frame	No

5.4.2 Flexibility in complexity cost modelling

Depending on how the decoder is implemented, there are different ways to model the complexity cost. Table 5.4, Table 5.5 and Table 5.6 list the results for models defined in Equation (5.13), Equation (5.14) and Table 5.2 respectively. Complexity cost values are normalized for comparison convenience. Sequence $Foreman$, $Stefan$ and $Mobile$ at $400Kbps$ with baseline profile are used. The quality degradation and complexity saving are relative to original JM82 results. Several facts can be observed from the data:

1. In general these results show similar R-D-C performance: a considerable ratio of interpolation computation can be saved with relatively much smaller quality degradation. This demonstrates the flexibility and robustness of CAMED for different complexity modelling methods.

2. The model defined in Table 5.2 has less saving ratio compared with others. This is because lots of computational cost has already been saved by efficient decoder design. Whereas its performance is still statistically meaningful and promising (see the results for $Foreman$ and $Stefan$).

3. Using the same value of Lagrange multiplier CAMED has different saving ratio for various content, which is a natural indicator on how difficult it is to replace H.264 motion information using low-complexity alternatives. E.g., the characteristic of sequence $Mobile$ (const camera panning makes the SKIP mode in P frame very efficient and hard to be beaten) explains the reason why it has less complexity saving ration compared to $Foreman$ and $Stefan$. This can be compensated using a larger Lagrange multiplier.

Without losing generality, in the remaining part of this chapter the model defined in Equation (5.13) is adopted. Though the generated bit streams (mainly the selected motion vectors and block modes) using other models might be different, similar performance results are observed through our experiment.

Table 5.4: CAMED performance using Equation (5.13)

γ_{MODE}		5	10	25	50
Foreman	Quality Degradation (dB)	0.091	0.147	0.251	0.372
	Complexity Saving (%)	63.38	70.08	80.24	85.63
Stefan	Quality Degradation	0.021	09.060	0.164	0.214
	Complexity Saving (%)	22.23	28.01	36.61	44.68
Mobile	Quality Degradation	0.035	0.091	0.161	0.274
	Complexity Saving (%)	8.47	13.27	18.30	24.88

Table 5.5: CAMED performance using Equation (5.14)

γ_{MODE}		5	10	25	50
Foreman	Quality Degradation (dB)	0.101	0.139	0.262	0.395
	Complexity Saving (%)	55.01	56.06	71.25	81.20
Stefan	Quality Degradation	0.031	0.029	0.106	0.155
	Complexity Saving (%)	14.61	14.78	24.42	32.09
Mobile	Quality Degradation	0.006	0.037	0.097	0.183
	Complexity Saving (%)	4.12	5.15	7.85	11.77

Table 5.6: CAMED performance using Table 5.2

γ_{MODE}		5	10	25	50
Foreman	Quality Degradation (dB)	0.037	0.074	0.182	0.326
	Complexity Saving (%)	30.04	41.92	58.04	70.36
Stefan	Quality Degradation	0.011	0.012	0.043	0.104
	Complexity Saving (%)	6.50	8.57	14.93	18.82
Mobile	Quality Degradation	0.021	0.052	0.024	0.065
	Complexity Saving (%)	2.56	2.88	3.93	5.75

5.4.3 Rate-distortion-complexity performance in CAMED

Figure 5.11 lists the rate-distortion performance together with rate-complexity results by different γ_{MODE} values. The latter is measured in terms of the ratio of the remaining complexity when applying the proposed CAMED method to the original complexity when using the H.264 JM82 codec (i.e., $\gamma_{MODE} = 0$). Note the complexity includes only the interpolation computation required for reconstructing the reference signals in the decoder, which is the most demanding component in the decoding process.

Several important findings are in order. First, adjusting γ_{MODE} is an efficient way to control the computational complexity. Up to 95% of the interpolation cost can be removed within a relatively small range of γ_{MODE}(see $Foreman$ at $1000Kbps$ with $\gamma_{MODE} = 500$). Secondly, the video quality is well maintained when reducing the complexity. If we use 0.5dB as the perceptual quality difference threshold, up to 60% of the computational cost can be saved without visible impairment (see $Stefan$ at $1000Kbps$ with $\gamma_{MODE} = 50$ and PSNR drop of 0.197dB). Figure 5.12 further shows the frame-to-frame quality and complexity over the entire video sequence $Stefan$. In fact, for all sequences, $30 \sim 50\%$ cost saving can be obtained within 0.1dB quality loss. According to the benchmark provided in [38], this can be translated into an overall decoding complexity saving up to 30%[1].

The reason of the above excellent performance probably can be attributed to the statistical characteristic of video signals. Without theoretical explanation, our intuitive conjecture is that not every sub-pixel motion vector is equally important in predicting the reference signal required in the motion compensation process. Moving less critical ones to the alternatives with reduced complexity will not dramatically increase the prediction error, but will help significantly in reducing the computational cost at the decoder. Figure 5.13 [12] show the R-D joint cost distribution choosing different sub pixel locations. It is evident that around the optimal motion vectors there are many alternative options that yield similar R-D cost while having different complexity. This important feature, however, could not be utilized using R-D framework. Our proposed CAMED can efficiently take advantage of this. Figure 5.14 shows one example comparing the motion vector distribution with and

[1]Depending on the decoder implementation, this value may vary

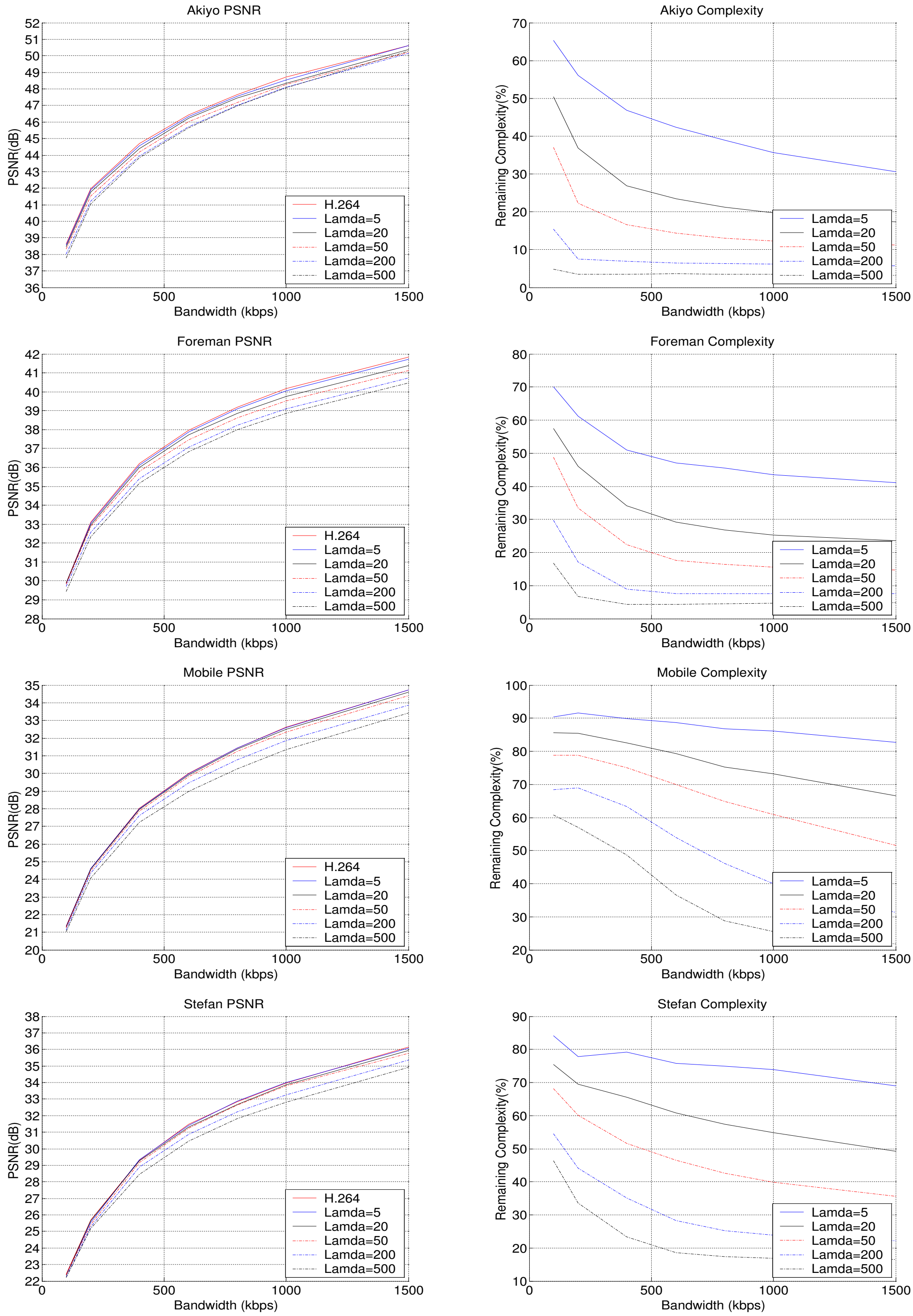


Figure 5.11: Performance of rate-distortion and rate-complexity using the proposed CAMED system.

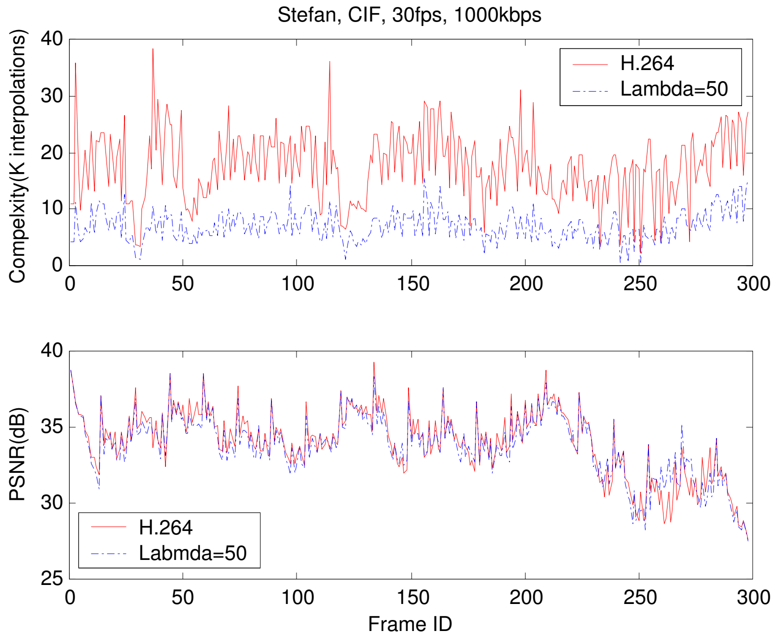


Figure 5.12: Frame-to-frame video quality and computational complexity comparison. The video quality is well maintained though the complexity is greatly reduced.

without applying the proposed CAMED method. It is obvious that many motion vectors shift to the locations with lower interpolation complexities.

5.4.4 Compatibility with fast motion estimation

Fast motion estimation (FME) is widely used in video encoding applications in order to reduce encoding complexity. In FME not all pixels (sub pixels) are checked, which might have potential influence on CAMED in that the optimal motion vector with best rate-distortion-complexity performance might be skipped. We also investigate the compatibility of our CAMED with FME. The experiment results in this section are obtained using baseline profile. FME is switched on for integer motion vectors, and subpixel motion estimation use either 3-step full search (FS) [104] or

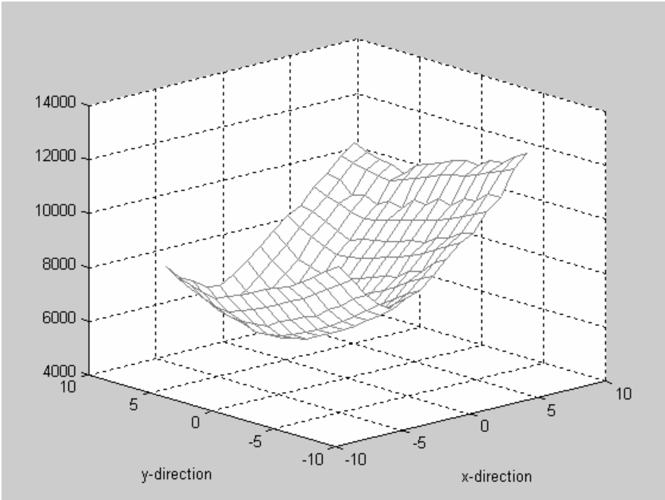


Figure 5.13: R-D joint cost distribution choosing different sub pixel locations.

fast fractional motion estimation (CBFPS, or center biased fractional pel search, the algorithm described in [12]).

Table 5.7 and Table 5.8 list the result for sequence *Foreman* and *Mobile* at $400Kbps$. Four results are compared: FS, CBFPS, CAMED, and CAMED+CBFPS. For the last one different Lagrange multipliers are adopted. From the results several conclusions can be drawn:

1. CBFPS can efficiently reduce the encoding complexity (in terms of the amount of checked sub pixels). Nevertheless, it has little contribution on decoding complexity (in terms of the amount of MC interpolation). Using other state-of-the-art FME algorithms [11, 103] we can get similar results. This is reasonable because according to their formulation these FME algorithms do not guarantee the complexity performance in the decoder side.

2. As in FS case CAMED can still reduce the decoding complexity efficiently with CBFPS presented. The quality degradation range after using CAMED is

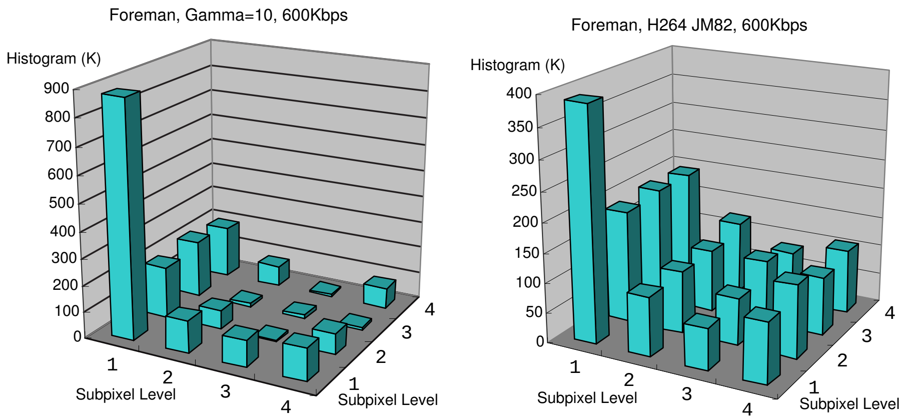


Figure 5.14: Subpixel motion vector distribution with and without the proposed CAMED method.

still within ignorable range (e.g., for *Foreman* 70.08% complexity saving with 0.147dB quality degradation). This demonstrates that CAMED has good compatibility with FME algorithms. Plus, the quality degradation after using both CBFPS and CAMED can be considered as additive from each of them, as further depicted in Figure 5.15.

3. It is very alluring to notice that by applying both CBFPS and CAMED, we can save the complexity from both encoder and decoder sides on purpose while keeping the video quality well maintained. We believe this is very practical for real applications to achieve encoding-decoding joint complexity reduction.

5.4.5 Complexity control

During complexity control, the basic operations are to adjust γ_{MODE} and manage the complexity buffer. First of all, the complexity model presented in Equation (5.20) need to be verified. Figure 5.9 illustrates the relationship between Lagrange multiplier and resulting complexity for the sequence $Mobile$ at $600Kbps$. Each curve

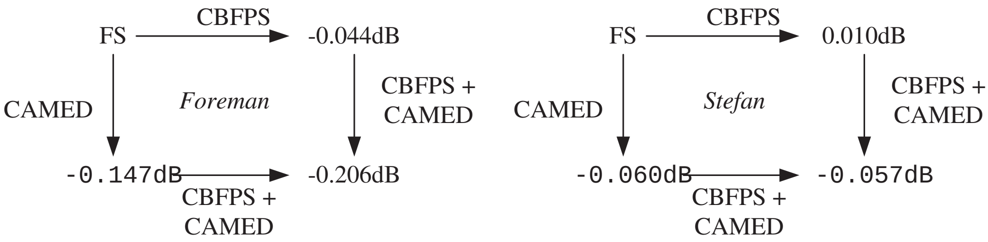


Figure 5.15: The quality degradation after using both CBFPS and CAMED can be considered as additive from each of them.

Table 5.7: Evaluation of compatibility with FME: *Foreman*

Foreman	PSNR(dB)[†]	Encoding Saving[♮]	Decoding Saving[♯]
FS	36.024	0.00%	0.00%
CBFPS	-0.044	61.80%	17.39%
CAMED $\gamma_{MODE} = 10$	-0.147	0.00%	70.08%
CBFPS+$\gamma_{MODE} = 10$	-0.206	64.75%	74.28%
CAMED $\gamma_{MODE} = 50$	-0.312	0.00%	85.63%
CBFPS+$\gamma_{MODE} = 50$	-0.390	66.09%	86.49%

[†] Except for FS, other values indicate the quality degradation compared with FS

[♮] Encoding saving is based on the amount of checked sub pixels

[♯] Decoding saving is based on the amount of interpolations

Table 5.8: Evaluation of compatibility with FME: *Stefan*

Stefan	PSNR(dB)	Encoding Saving	Decoding Saving
FS	29.422	0.00%	0.00%
CBFPS	0.01	61.68%	2.05%
CAMED $\gamma_{MODE} = 10$	-0.060	0.00%	28.01%
CBFPS+$\gamma_{MODE} = 10$	-0.057	62.58%	27.89%
CAMED $\gamma_{MODE} = 50$	-0.214	0.00%	44.67%
CBFPS+$\gamma_{MODE} = 50$	-0.253	63.45%	45.03%

is for one P or B frame. The linear relationship between the complexity and logarithmic γ_{MODE} is evident, especially for P frames. B frames usually have larger complexity because of bi-directional prediction. For the same frame type, we hypothesize the variation is caused by different content complexity in each frame. Like the empirical approach used in the conventional rate control process, we use MAD as an approximate measure of the frame content complexity. Figure 5.16 compares the frame-to-frame evolution of computational complexity with some major coding parameters for the P frames in the $Mobile$ sequence at $600Kbps$ with $\gamma_{MODE} = 5$. Compared to other options (such as quantization parameter), MAD appears to demonstrate the closest correlation with the computational though some variance is still noticeable. Study of improved measures capturing the statistical properties of the computational complexity is an interesting topic for future research.

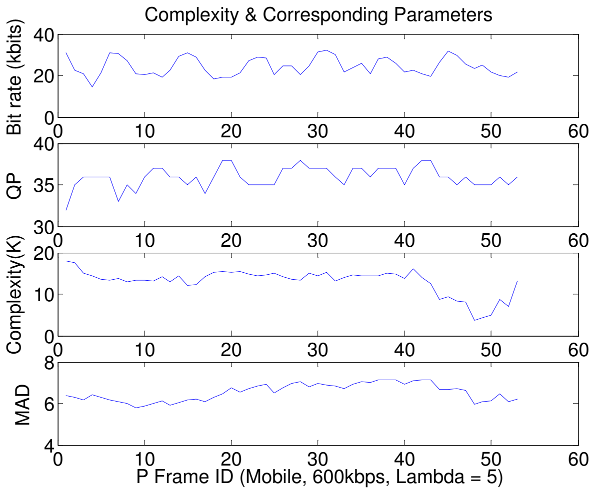


Figure 5.16: Relationship between computational complexity and major coding parameters.

Table 5.9 lists the main parameters used in the complexity control experiment. Some parameters may be further fine-tuned in order to find better tradeoff between quality fluctuation and control efficiency. We leave such potential improvement in future studies.

Figure 5.17 shows the detailed complexity control results for the sequence Foreman at $1000Kbps$ with different target complexity levels. The complexity of the

Table 5.9: Parameters used in complexity control

Basic control unit	One frame
Frame complexity prediction window size	6
Frame complexity prediction model	Linear
γ_{MODE} prediction window size	6
γ_{MODE} prediction model	Equation (5.21)
γ_{MODE} range	[0 5000]
Maximum γ_{MODE} change magnitude	80
Initial ratio of B/P frame complexity	1.6
Initial γ_{MODE} value	10
Target complexity per GOP (K interpolations)	$50K \sim 250K$

baseline H.264 JM82 result is also shown for comparison. The results are very promising. Though in all cases the initial γ_{MODE} is set to the same value 10, it can be adaptively adjusted and the target complexity level is consistently accomplished, except for the case aggressively reducing the complexity from about 250K to 50K (80% reduction). The latter aggressive case will not be achievable without sacrificing greatly the video quality. For other cases, some fluctuation can still be seen at the end of the sequence starting from 14^{th} GOP, where the sequence contains rapid camera panning. Our complexity control method cannot completely smooth this huge complexity increase because we bound the maximum magnitude and the maximum change rate of the γ_{MODE} parameter in order to avoid excessive quality loss and quality fluctuation. In other words, we try to maintain some consistence in video quality throughout the entire video sequence as well.

Table 5.10 summarizes the complexity control performance at $1000Kbps$. Complexity control error is calculated as the difference between the actual resulting complexity and the target complexity, normalized by the target complexity. Complexity Saving is the percentage of the original computational cost that has been removed. Quality Degradation is the quality difference (in PSNR) between the bit stream

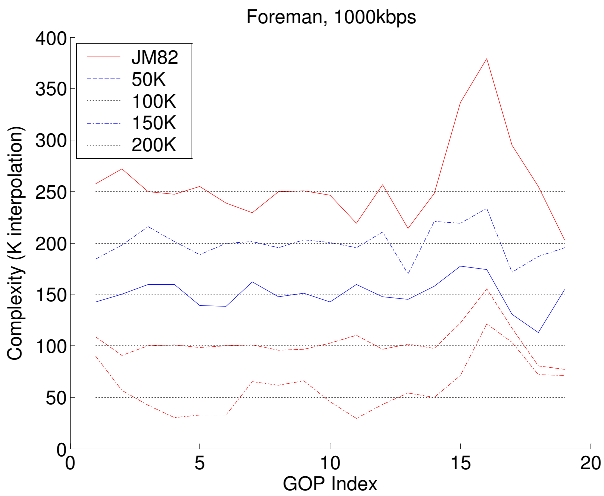


Figure 5.17: Complexity control performance.

generated using original H.264 and the one using our complexity control method. These results confirm that large savings of the computational complexity (30% to 60%) can be achieved with small quality degradation (0.3dB). Improvements from different video clips are different depending on the type of the content and the complexity of the signal. The most challenging case is the *Mobile* sequence, which has a steady camera motion (slowly panning left) and thus the SKIP/DIRECT mode is frequently used. It is difficult for the proposed CAMED method to change the motion vector to the integer one without incurring significant increase of bit rate. This is partly because that the SKIP/DIRECT mode is already a very efficient coding mode - very few bits are needed in coding the mode information and changing the mode will not improve much the prediction accuracy due to the motion in the video content. However, even for such a challenging case, our proposed CAMED method can still achieve about 33% complexity saving in order to keep the video quality more or less intact.

Table 5.11 summarizes the performance of complexity control at $100Kbps$ with

Table 5.10: Complexity control performance (1000$Kbps$)

Target Complexity (6-tap Interpolation)		**50K**	**100K**	**150K**	**200K**	**250K**
Foreman	Complexity control error (%)	19.78	2.81	0.09	0.2	1.4
	Complexity saving (%)	76.78	60.15	41.80	22.63	4.45
	Quality degradation (dB)	0.60	0.28	0.13	0.06	0.02
Stefan	Complexity control error (%)	61.43	1.73	1.16	0.14	4.37
	Complexity saving (%)	67.68	59.27	39.25	20.04	4.28
	Quality degradation (dB)	0.78	0.60	0.35	0.05	-0.09
Mobile	Complexity control error (%)	195.8	47.92	1.37	5.24	11.59
	Complexity saving (%)	47.92	47.92	47.91	33.27	22.18
	Quality degradation (dB)	0.63	0.63	0.63	0.31	0.13

baseline profile. It is clear to observe that the performance at the lower bit rate is better than that at the higher bit rate. The reason is that at lower bit rate the motion information is less sensitive because the residual errors are rough (i.e., with less entropy values) due to severe quantization. In another words it is more reliable to reduce the MC complexity with trivial quality degradation by using R-D-C optimized motion information. On the other hand, the motion information adopted in B frames is more difficult to be beaten using CAMED because of the highly efficient temporal direct mode [86]. Since B frames are not applied to baseline profile, this benefits the usage of CAMED. Typical power aware video decoding scenarios (such as mobile/wireless video applications) involve low bitrate coding using baseline profile. Our results indicate that CAMED is an excellent solution for such applications.

5.5 Summarization

In this chapter we present a novel complexity adaptive motion estimation and mode decision method (CAMED) and apply it on the emerging coding standard H.264 so that the computational complexity required at the decoder can be greatly reduced

Table 5.11: Complexity control performance (100$Kbps$)

Target Complexity (6-tap Interpolation)		**5K**	**10K**	**20K**	**30K**	**40K**
Foreman	Complexity control error (%)	37.07	5.00	1.50	3.38	3.09
	Complexity saving (%)	88.06	81.70	65.67	45.95	28.14
	Quality degradation (dB)	0.22	0.16	0.09	0.03	0.03
Stefan	Complexity control error (%)	39.38	19.45	0.32	0.26	0.37
	Complexity saving (%)	88.07	86.21	65.64	48.49	31.25
	Quality degradation (dB)	0.40	0.30	0.09	-0.03	0.06
Mobile	Complexity control error (%)	114.97	7.48	0.65	1.34	4.73
	Complexity saving (%)	81.89	81.89	66.08	50.13	35.79
	Quality degradation (dB)	0.40	0.40	0.17	0.14	0.06

while the video quality is maintained with very little degradation. Such results are very useful for the increasingly popular handheld devices in many mobile applications.

We first analyze the decoding complexity behavior and identify the most critical components, i.e., the motion vector and the block mode that affect the cost of the interpolation process in motion compensation. We develop simple but practical cost functions to estimate the required computation for each motion vector and block mode. Both platform-independent and platform-dependent models are discussed. Thereafter we extend the conventional rate-distortion optimization framework based on the Lagrange multiplier method to explicitly handle the computational complexity. In addition, for complexity control, we propose an effective logarithmic-linear model to predict the relationship between the target complexity and the Lagrange multiplier. The joint rate-distortion-complexity framework together with the complexity control algorithm provides an effective solution for optimizing the tradeoff between video quality and resources, including both bit rate and computational complexity. The proposed system can be easily embedded into existing video coding systems and will work with any standard compatible decoder.

Our extensive experiments using H.264 codec over different video sequences, different bit rates, and different complexity levels demonstrate the power of the proposed system. Up to 60% of the interpolation complexity can be saved at the decoder without incurring noticeable quality loss (within 0.2 dB). Even for challenging video clips such as *Mobile*, 33% of the complexity can be reduced with quality difference less than 0.3dB. The proposed complexity control scheme can reliably meet the target complexity requirement for a wide range of video content.

Chapter 6

Conclusion and Future Work

6.1 Thesis summarization

In this thesis we formulate a generic resource constrained video coding/adaptation problem by extending the conventional rate-distortion (R-D) framework. Various resource constraints, coding parameters (adaptation operations), and quality evaluation metrics are considered for this problem. Specifically we investigate two important issues: utility function (UF) based multiple dimensional adaptation (MDA) with content based operation prediction, and complexity adaptive H.264 decoding.

6.1.1 UF based MDA with content based operation prediction

We propose a new adaptation framework based on utility function which is defined in the adaptation-resource-utility (ARU) space. We describe the usability of UF in guiding the selection of MDA operations. UF is one of the formats supported by AdaptationQoS in MPEG-21 DIA in providing adaptation metadata information. To address the issue of UF generation for real time applications, we present the content-based operation prediction framework. Instead of conducting analytical modelling, a statistical based system is constructed where low level content features from the compress domain are employed to predict the UF information through machine learning methods. In order to testify the performance of the proposed framework, we instantiate our system using two typical video codecs: non-scalable codec MPEG-4 and scalable codec motion compensated embedded zeroblock coding (MC-EZBC).

SNR-temporal MDA behavior for MPEG-4 codec is formed using frame dropping (FD) and coefficient dropping (CD) combined adaptation. We uniquely define FD and CD operations so that a wide range of bit rate can be covered. To address the FD-CD rate control issue, two approaches are proposed: one is the compensation of bit drift caused by VLC symbol re-encoding; another is the buffer management with FD presented. The experiment results verify the excellent performance of our methods. Thereafter, UF representation of FD-CD is described by collecting a set of adaptation anchor nodes that are specific combination of FD-CD operations. To apply the proposed content based prediction framework on FD-CD, the prediction model is carefully derived and the framework constituting unsupervised clustering, supervised classification and regression is generated. The extensive experiment results demonstrate very promising prediction accuracy (up to 89%) over diverse types of video content.

MC-EZBC is one of the latest scalable video codec. The SNR-spatio-temporal scalability of MC-EZBC is analyzed, and SNR-temporal combined MDA is constructed using both bitplane and temporal subband truncation. In contrast with the objective SNR metrics used in MPEG-4 FD-CD, we launch thorough subjective experiment using a large pool of diverse video data (128 video clips) and a modest group of (31) human subjects. Formal statistical analysis is conducted to assess the statistical significance of the experiment. Based on the experiment data, we discover distinctive patterns of subjective preferences of different SNR-temporal resolutions when adapting video under different resource (bitrate) constraints. The generated subjective UF can be adopted to select the optimal MDA operation efficiently. The content-based prediction framework is applied to MC-EZBC and the performance is evaluated. The experiment results indicate that the prediction system can effectively reveal the relationship between the content characteristic and the MDA behavior, and therefore accurately predict the optimal adaptation operation with accuracy from 77% to 95% over different bandwidth. In addition, we investigate the feature selection issue to identify the optimal set of content features leading to good balance between computing complexity and classification accuracy.

6.1.2 Complexity Adaptive Motion Estimation and Mode Decision for H.264 Video

Emerging video coding standard H.264 achieves considerably improved video quality compared to its predecessors at the cost of significantly increased computational complexity. We propose a novel complexity adaptive motion estimation and mode decision (CAMED) system to optimize the selection of the motion vectors and motion compensation block modes in order to dramatically reduce the decoding computational cost while keeping the video quality well maintained. Our method can be applied to any existing H.264 encoder system and is compatible with any standard-compliant decoder. Technically we accomplish this goal by applying the following methods:

1. Applying a rigorous methodology to extend the conventional rate-distortion optimization framework to include the computation (C) term.

2. Estimating the complexity cost functions in different aspects, such as platform-independent and platform-dependent modelling, as well as hardware-based

modelling.

3. Developing a complexity model that can reliably determine the appropriate parameter (i.e., Lagrange multiplier) needed for optimizing the R-D-C tradeoff relationships.

4. Designing a complexity-control algorithm to meet the specified target complexity level while maintaining the video quality.

Our extensive experiments with different video contents, bit rates, and complexity levels show very satisfying results in reducing the number of interpolation by up to 60% while keeping the video quality almost intact (quality degradation less than 0.2dB). The proposed complexity control scheme can reliably meet the target complexity requirement for a wide range of video content. We believe the usage of the CAMED on mobile/wireless video applications is very promising.

6.2 Open issues and future work

We have discussed some open issues and future direction in each of the chapters presented earlier. A quick summary is provided as follows.

1. We have developed a generic resource constrained video coding framework is proposed in this thesis to accommodate diverse types of operation parameters, resource constraints, and quality requirement. We have shown how to model the relationships (utility function) among operations, resources, and quality, through several examples using actual coding systems. Such relationships are important for guiding the optimal selection of adaptation operation. However, given new types of resources and operations, it remains as a challenging issue to find appropriate quantitative representation and/or mathematical models

for the relationships. Such representations and models are needed in the later process of finding the optimal adaptation. We further discuss some specific issues below.

- In video adaptation, objective quality metric such as SNR is suitable for quality measurement when only SNR adaptation is applied. In MDA applications, however, it is still difficult to find an efficient metric to unify the quality degradation from different adaptation dimensions involving spatial, temporal, and resolution variations.
- We have addressed SNR-temporal combined MDA adaptation in this thesis. However, we have not incorporated spatial scalability and expand the MDA to a joint SNR-spatial-temporal one. A full SNR-spatio-temporal MDA is very important for applications in universal media access (UMA) scenario. Therefore deeper understanding on their behaviors and optimal operation selection issue is worth further investigation.
- A empirical approach for complexity modelling has been used in CAMED for achieving effective complexity control. In our work, empirical statistical data has been adopted to derive an approximate model. More rigorous theoretical analysis on the relationship between the complexity level and the control parameter (i.e., the Lagrange multiplier) will be worthwhile in future work.

2. The content based prediction framework employs signal-level content features (such as motion vectors and texture energy) from the compressed domain, resulting in accuracy in predicting the optimal MDA operation. In fact high level semantic features (such as camera/object motion understanding and video genre) are also very important for describing characteristic of the video con-

tent, which in turn facilitates the prediction procedure. Nevertheless extracting these features from the bit streams may require sophisticated computation. How to efficiently retrieve such semantic features in a light-weight manner and fuse them into the prediction framework is a promising while challenging topic.

3. CAMED has great potential in realizing efficient low-complexity video decoding for H.264. There are several interesting topics for further exploration. First, in practice, many video decoders leverage special hardware features (such as MMX and SIMD) to reduce the decoding complexity. Also other implementation details (such as memory access strategy and system-level design issues) will substantially change the complexity breakdown of different components in the decoder. It is an important practical topic to study how the proposed technique will affect the video quality and computational complexity when such implementation details are considered. Second, several components of the proposed framework, such as the complexity modelling, are not fully optimized and might be dependent on the selection of specific complexity cost functions. These present interesting opportunities for further improvement. Finally, in complexity control, the resource limitation is the target complexity level, which is difficult to be obtained in real system. Instead, an alternative approach is to apply the complexity control based on certain quality degradation constraint (e.g., no larger than 0.3dB). Specific solutions for this issue is worthy of further exploration.

Bibliography

[1] Y. Andreopoulos, M. van der Schaar, A. Munteanu, J. Barbarien, P. Schelkens, and J. Cornelis, “Fully-scalable wavelet video coding using in-band motion-compensated temporal filtering,” in *Proc. Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 3, March 2003, pp. 471–420.

[2] D. Barbara, J. Couto, , and Y. Li, “Coolcat: An entropy-based algorithm for categorical clustering,” in *In Conference on Information and Knowledge Management (CIKM)*, 2002.

[3] R. Battiti, “Using mutual information for selecting features in supervised neural-net learning,” *IEEE Trans. Neural Network*, vol. 5, no. 4, pp. 537–550, 1994.

[4] N. Björk and C. Christopoulos, “Video transcoding for universal multimedia access,” in *ACM Multimedia Workshops*, 2000, pp. 75–79.

[5] P. Bocheck, A. T. Campbell, S.-F. Chang, and R.-F. Liao, “Content-aware network adaptation for mpeg-4,” in *ACM International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, June 1999.

[6] P. Bocheck, Y. Nakajima, and S.-F. Chang, “Realtime estimation of subjective utility functions for mpeg-4 video objects,” in *Proceedings of IEEE Packet Video Workshop (PV’99)*, Apr. 1999.

[7] S.-F. Chang, “Optimal video adaptation and skimming using a utility-based framework,” in *Proc. Int. Work. Digital Comm.*, Capri Island, Italy, Sep. 2002.

[8] S.-F. Chang and D. G. Messerschmitt, “A new approach to decoding and compositing motion compensated dct-based images,” in *Proc. IEEE Intern. Conf. on Acoustics, Speech, and Signal Processing*, 1993, pp. V421–V424.

[9] P. Chen and J. W. Woods, "Bidirectional mc-ezbc with lifting implementation," *IEEE Trans. Circuits Syst. Video Techn.*, vol. 14, no. 10, pp. 1183–1194, 2004.

[10] T.-C. Chen, Y.-C. Huang, and L.-G. Chen, "Full utilized and resuable architecture for fractional motion estimation of h.264/avc," in *Proc. Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, May 17-21, 2004.

[11] Z. Chen and Y. He, "Prediction based directional refinement (pdr) algorithm for fractional pixel motion search strategy," in *JVT-D069, 4th meeting*, Klagenfurt, Austria, 22-26 July, 2002.

[12] Z. Chen, P. Zhou, and Y. He, "Fast integer pel and fractional pel motion estimation for jvt," in *Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVT-F017*, Awaji, Japan, 5-13 Dec. 2002.

[13] H.-Y. Cheong and A. M. Tourapis, "Fast motion estimation within the h.264 codec," in *Proc. Intl. Conf. on Multimedia and Expo (ICME)*, July 6-9, 2003.

[14] T. Chiang and Y.-Q. Zhang, "A new rate control scheme using quadratic rate distortion model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 246–250, Feb. 1997.

[15] S.-J. Choi and J. W. Woods, "Motion-compensated 3-d subband coding of video," *IEEE Trans. on Image Processing*, vol. 8, no. 2, pp. 155–167, February 1999.

[16] C. Chou and Y. Li, "A perceptually tuned subband image coder based on the measure of just-noticeable-distortion profile," *IEEE Trans. Circuits Systems Video Technol.*, vol. 5, no. 6, pp. 467–476, December 1995.

[17] C. Christopoulos, A. Skodras, and T. Ebrahimi, "The jpeg2000 still image coding system: an overview," *Consumer Electronics, IEEE Transactions on*, vol. 46, no. 4, pp. 1103 –1127, Nov. 2000.

[18] T. Chu, S. Sun, L. Kerofsky, and S. Lei, "Discussion on efficient 16-bit implementation of the interpolation routine," in *ITU-T SG16 Doc. VCEG-N19*, 2001.

[19] Q. Dai and J. Wu, "Construction of power efficient routing tree for ad hoc wireless networks using directional antenna," in *Proc. International Conference on Distributed Computing Systems Workshops*, 23-24 Mar. 2004, pp. 718–722.

[20] K. deok Seo, S. kak Kwon, S. K. Hong, and J. kyoon Kim, “Dynamic bit-rate reduction based on frame-skipping and requantization for mpeg-1 to mpeg-4 transcoder,” in *Proc. International Symposium on Circuits and Systems (ISCAS)*, vol. 2, 25-28 May 2003, pp. 372–375.

[21] Y. Eisenberg, C. E. Luna, T. N. Pappas, R. Berry, and A. Katsaggelos, “Joint source coding and transmission power management for energy efficient wireless video communications,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 6, pp. 411–424, June 2002.

[22] A. Eleftheriadis, “Dynamic rate shaping of compressed digital video,” Ph.D. dissertation, Graduate School of Arts and Sciences, Columbia University, June 1995.

[23] M. Ellis, M. V. Spakovsky, and D. Nelson, “Fuel cell systems: efficient, flexible energy conversion for the 21st century,” *Proceedings of the IEEE*, vol. 89, no. 12, pp. 1808 – 1818, Dec. 2001.

[24] K.-T. Fung, Y.-L. Chan, and W.-C. Siu, “New architecture for dynamic frame-skipping transcoder,” *IEEE Transactions on Image Processing*, vol. 11, no. 8, pp. 886–900, 2002.

[25] B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero, “Distributed video coding,” *Proc. of the IEEE, Special Issue on Video Coding and Delivery*, vol. 93, no. 1, pp. 71–83, 2005.

[26] R. Gonzalez and R. Woods, *Digital Image Processing (2nd edition)*. New York: Prentice Hall, 2002.

[27] D. Gotz and K. Mayer-Patel, “A general framework for multidimensional adaptation,” in *Proceedings of the 12th annual ACM international conference on Multimedia (ACM MM’04)*. New York, NY, USA: ACM Press, 2004, pp. 612–619.

[28] H.-M. Hang and J.-J. Chen, “Source model for transform video code and its application, part i and ii,” *IEEE Trans. Circuits Syst.*, vol. 7, no. 2, pp. 287–311, Apr. 1997.

[29] Z. He, Y. Liang, L. Chen, I. Ahmad, and D. Wu, “Power-rate-distortion analysis for wireless video communication under energy constraints,” *IEEE Trans. Circuits Syst. Video Technol., Special Issue on Integrated Multimedia Platforms*, vol. 15, no. 5, pp. 645–658, May 2005.

[30] S.-T. Hsiang and J. W. Woods, "Embedded video coding using motion compensated 3-d subband/wavelet filter bank," in *Packet Video Workshop*, May 2000.

[31] S.-T. Hsiang and J. Woods, "Highly scalable subband/wavelet image and video coding," Ph.D. dissertation, Rensselaer Polytechnic Institute, Troy, NY, May 2002.

[32] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multi-class support vector machines," *IEEE Transactions on Neural Networks*, vol. 13, p. 415C425, 2002.

[33] A. Islam and W. A. Pearlman, "An embedded and efficient low-complexity hierarchical image coder," in *Proc. SPIE Visual Communications and Image Processing (VCIP)*, Jan. 1999, pp. 294–305.

[34] Z. Ji, Q. Zhang, W. Zhu, and Y.-Q. Zhang, "End-to-end power-optimized video communication over wireless channels," in *Proc. of IEEE Workshop on MMSP*, 2001.

[35] H. Kim and Y. Altunbasak, "Low-complexity macroblock mode selection for the h.264/avc encoders," in *IEEE Int. Conf. on Image Processing (ICIP)*, October 2004.

[36] J.-G. Kim, Y. Wang, S.-F. Chang, and H.-M. Kim, "An optimal framework of video adaptation and its application to rate adaptation transcoding," *ETRI Journal*, vol. 27, no. 4, August 2005.

[37] R. Koenen, "Overview of the mpeg-4 standard," in *ISO/IEC JTC1/SC29/WG11 N3536*, July 2000.

[38] V. Lappalainen, A. Hallapuro, and T. Hamalainen, "Complexity of optimized h.26l video decoder implementation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 717–725, Jul. 2003.

[39] H.-J. Lee, T. Chiang, and Y.-Q. Zhang, "Scalable rate control for very low bit rate (vlbr) video," in *Proc. International Conference on Image Processing (ISCAS)*, vol. 2, Washington, DC, USA, October 26 - 29, 1997, pp. 768–771.

[40] J. Lee, S. Moon, and W. Sung, "H.264 decoder optimization exploiting simd instructions," in *IEEE Asia-Pacific Conference on Circuits and Systems, (APCCAS)*, vol. 2, December 2004, pp. 1149–1152.

[41] Z. Lei and N. Georganas, “H.263 video transcoding for spatial resolution downscaling,” in *Proc. Intl. Conference on Information Technology: Coding and Computing*, 8-10 April 2002, pp. 425–430.

[42] K. Lengwehasatit and A. Ortega, “Rate complexity distortion optimization for quadtree-based dct coding,” in *IEEE Int. Conf. on Image Processing (ICIP)*, September 2000.

[43] Y. Liang and Y.-P. Tan, “A new content-based hybrid video transcoding method,” in *IEEE Int. Conf. on Image Processing (ICIP)*, October 2001, pp. 429–432.

[44] Y. Liu and J. R. Kender, “Fast video segment retrieval by sort-merge feature selection, boundary refinement, and lazy evaluation,” *Computer Vision and Image Understanding*, vol. 92, no. 2-3, pp. 147–175, November-December 2003.

[45] R. Lowry, “Concepts and applications of inferential statistics.” [Online]. Available: http://faculty.vassar.edu/lowry/webtext.html

[46] X. Lu, E. Erkip, Y. Wang, and D. Goodman, “Power efficient multimedia communication over wireless channels,” *IEEE Journal on Selected Areas on Communications, Special Issue on Recent Advances in Wireless Multimedia*, vol. 21, no. 10, pp. 1738–1751, Dec. 2003.

[47] M. A. Masry and S. S. Hemami, “Cvqe: A continuous video quality evaluation metric for low bit rates,” in *Proc. SPIE Human Vision and Electronic Imaging*, January 2003.

[48] Microsoft, “Microsoft mpeg-4 reference model.” [Online]. Available: http://standards.iso.org/ittf/PubliclyAvailableStandards/ISO_IEC_14496-5_2001_Software_Reference/

[49] R. Mohan, J. R. Smith, and C.-S. Li, “Adapting multimedia internet content for universal access,” *IEEE Transactions on Multimedia*, vol. 1, no. 1, pp. 104–114, 1999.

[50] MoMuSys, “Momusys mpeg-4 reference model.” [Online]. Available: http://www.tnt.uni-hannover.de/project/eu/momusys/mom-overview.html

[51] D. G. Morrison, M. E. Nilsson, and M. Ghanbari, “Reduction of the bit-rate of compressed video while in its coded form,” in *Proc. Packet Video Workshop*, 1994, p. D17.1CD17.4.

[52] D. Mukherjee, E. Delfosse, J.-G. Kim, and Y. Wang, "Optimal adaptation decision-taking for terminal and network quality-of-service," *IEEE Transactions on Multimedia*, vol. 7, no. 3, pp. 454–462, Jun. 2005.

[53] J.-R. Ohm, "Three dimensional subband coding with motion compensation," *IEEE Trans. on Image Processing*, vol. 3, no. 5, pp. 559–571, September 1994.

[54] J. Platt, N. Cristianini, and J. Shawe-Taylor, "Large Margin DAGs for Multiclass Classification," *Advances in Neural Information Processing Systems*, vol. 12, pp. 547–553, 2000.

[55] W. K. Pratt, *Digital Image Processing (3rd edition)*. New York: Wiley-Interscience, July 24, 2001.

[56] R. Puri and K. Ramchandran, "Prism: a new robust video coding architecture based on distributed compression principles," in *Allerton Conf. Communication, Control, and Computing*, 2002.

[57] H. Radha, M. van der Schaar, and Y. Chen, "The mpeg-4 fine-grained scalable video coding method for multimedia streaming over ip," *IEEE Trans. on Multimedia*, vol. 3, no. 1, pp. 53–68, March 2001.

[58] R. K. Rajendran, M. van der Schaar, and S.-F. Chang, "Fgs+: Optimizing the joint spatio-temporal video quality in mpeg-4 fine grained scalable coding," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2002.

[59] R. L. Rardin, *Optimization in Operations Research*. New York: Prentice Hall, 1998.

[60] A. Ray and H. Radha, "Complexity-distortion analysis of h.264/jvt decoder on mobile devices," in *Picture Coding Symposium (PCS)*, December 2004.

[61] E. C. Reed and J. S. Lim, "Optimal multidimensional bit-rate control for video communications," *IEEE Trans. Image Processing*, vol. 11, no. 8, pp. 873–885, Aug. 2002.

[62] A. Rohaly, J. Libert, P. Corriveau, A. Webster, and et al., "Final report from the video quality experts group on the validation of objective models of video quality assessment," March 2000. [Online]. Available: www.vqeg.org

[63] A. Said and W. A. Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 243–250, June 1996.

[64] M. Schaar and H. Radha, "Adaptive motion-compensation fine- granular-scalability (amc-fgs) for wireless video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 6, pp. 360–371, 2002.

[65] N. Semiconductor, "National's powerwise technology." [Online]. Available: http://www.national.com/appinfo/power/powerwise.html

[66] T. Shanableh and M. Ghanbari, "Heterogeneous video transcoding to lower spatio-temporal resolutions and different encoding formats," *IEEE Transactions on Multimedia*, vol. 2, no. 2, pp. 101–110, June 2000.

[67] J. M. Shaprio, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3445–3462, December 1993.

[68] H. Sorial, W. Lynch, and A. Vincent, "Estimating laplacian parameters of dct coefficients for requantization in the transcoding of mpeg-2 video," in *Proc. International Conference on Image Processing (ICIP)*, BC, Canada, September 2000.

[69] Y. Su and M.-T. Sun, "Fast multiple reference frame motion estimation for h.264," in *Proc. IEEE International Conference on Multimedia and Expo (ICME)*, vol. 1, 27-30 June 2004, pp. 695–698.

[70] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 74–90, Nov. 1998.

[71] G. Sullivan, P. Topiwala, and A. Luthra, "The h.264/avc advanced video coding standard: Overview and introduction to the fidelity range extensions," in *SPIE Conference on Applications of Digital Image Processing*, Aug. 2004.

[72] H. Sun, W. Kwok, and J. Zdepski, "Architecture for mpeg compressed bitstream scaling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 191–199, 1996.

[73] H. Sundaram, L. Xie, and S.-F. Chang, "A utility framework for the automatic generation of audio-visual skims," in *ACM Multimedia*, Juan Les Pins, France, December 2002.

[74] D. S. Taubman, "High performance scalable image compression with ebcot," *IEEE Trans. Image Proc.*, vol. 9, pp. 1158–1170, July 2000.

[75] ARM and National Semiconductor, "Powering next generation mobile devices." [Online]. Available: http://www.national.com/appinfo/power/files/powering_next_gen_mobile_devices.pdf

[76] ISO/IEC, "Report on 3dav exploration," in *ISO/IEC JTC1/SC29/WG11 N5878*, July 2003.

[77] ——, "Description of core experiments in mpeg-21 scalable video coding," in *ISO/IEC JTC1/SC29/WG11 N6521*, Redmond, WA, July 2004.

[78] ——, "Coding of audio-visual objects: Visual," in *ISO/IEC JTC1/SC29/WG11 N250 14496-2*, November 1998.

[79] ——, "Study of iso/iec 21000-7 fcd - part 7: Digital item adaptation," in *ISO/IEC JTC1/SC29/WG11/N5933*, Oct. 2003.

[80] ——, "Mpeg-21 overview v.5," in *ISO/IEC JTC1/SC29/WG11/N5231*, October 2002. [Online]. Available: http://mpeg.telecomitalialab.com/standards/mpeg-21/mpeg-21.htm

[81] ITU-T, "Methodology for the subjective assessment of the quality of television pictures," in *ITU Telecom. Standardization Sector, Recommendation ITU-R BT.500-10*, August, 2000.

[82] ——, "Video codec for audiovisual services at $p \times 64kbit/s$," in *ITU Telecom. Standardization Sector, Recommendation H.261*, Mar 1993.

[83] ITU-T and ISO/IEC, "Scalable video coding - working draft 1," in *Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG(ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6)*, Hong Kong, CN, 17-21 January, 2005.

[84] ——, "Draft itu-t recommendation and final draft international standard of joint video specification (itu-t rec. h.264 — iso/iec 14496-10 avc)," in *Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVT-G050*, March 2003.

[85] A. M. Tourapis, "Enhanced predictive zonal search for single and multiple frame motion estimation," in *Proceedings of Visual Communications and Image Processing(VCIP)*, Jan 2002, pp. 1069–1079.

[86] A. M. Tourapis, F. Wu, and S. Li, "Direct mode coding for bi-predictive pictures in the jvt standard," in *Proc. Intl. Symposium on Circuits and Systems(ISCAS)*, vol. 2, 2003, pp. 700–703.

[87] C. van den Branden Lambrecht, "Perceptual models and architectures for video coding applications," Ph.D. dissertation, Ecole Polytechnique Federale de Lousanne, Lausanne, EPFL, 1996.

[88] V. Vapnik, *Statistical Learning Theory.* New York: Wiley, 1998.

[89] A. Vetro, C. Christopoulos, and H. Sun, "Video transcoding architectures and techniques: An overview," *IEEE Signal Processing Magazine*, vol. 20, no. 2, pp. 18–29, March 2003.

[90] A. Vetro and C. Timmerer, "Digital item adaptation: Overview of standardization and research activities," *IEEE Transactions on Multimedia*, vol. 7, no. 3, pp. 418–426, June 2005.

[91] A. Vetro, Y. Wang, and H. Sun, "Rate-distortion optimized video coding considering frameskip," in *Proc. Intl. Conf. Image Processing (ICIP)*, Vancouver, Candada, 7-10 Oct. 2001, pp. 534–537.

[92] Y. Wang, S.-F. Chang, and A. C. Loui, "Subjective preference of spatio-temporal rate in video adaptation using multi-dimensional scalable coding," in *Proc. Intl. Conf. on Multimedia and Expo (ICME), special session on Mobile Imaging: technology and applications*, vol. 3, June 27-30, 2004, pp. 1719–1722.

[93] Y. Wang, J. Ostermann, and Y.-Q. Zhang, *Video Processing and Communications.* New York: Prentice Hall, September 2001.

[94] Y. Wang, J.-G. Kim, and S.-F. Chang, "Mpeg-4 real time fd-cd transcoding," Columbia University DVMM Group, Tech. Rep. 208-2005-2, 2003.

[95] ——, "Content-based utility function prediction for real-time mpeg-4 video transcoding," in *Proc. Intl. Conf. Image Processing (ICIP)*, vol. 1, 14-17 Sep. 2003, pp. 189–192.

[96] Y. Wang, T.-T. Ng, M. van der Schaar, and S.-F. Chang, "Predicting optimal operation of mc-3dsbc multi-dimensional scalable video coding using subjective quality measurement," in *Proc. SPIE Video Comm. and Image Processing (VCIP)*, 2004.

[97] A. B. Watson, J. H. J., and J. F. McGowan, "Dvq: A digital video quality metric based on human vision," *Journal of Electronic Imaging*, vol. 10, no. 1, pp. 20–29, 2001.

[98] T. Wedi and H. Musmann, "Motion- and aliasing-compensated prediction for hybrid video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 577–586, Jul. 2003.

[99] O. Werner, "Requantization for transcoding of mpeg-2 intraframes," *IEEE Trans. Image Processing*, vol. 8, pp. 179–191, Feb. 1999.

[100] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the h.264/avc video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[101] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "On the construction of energy-efficient broadcast and multicast trees in wireless networks," in *Proc. Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 2, 26-30 March 2000, pp. 585–594.

[102] G. Yadavalli, M. Masry, and S. Hemami, "Frame rate preferences in low bit rate video," in *IEEE Intl. Conf. on Image Processing (ICIP)*, 2003.

[103] L. Yang, K. Yu, J. Li, , and S. Li, "Prediction-based directional fractional pixel motion estimation for h.264 video coding," in *Proc.IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 18-23 March, 2005.

[104] P. Yin, A. M. Tourapis, H.-Y. Cheong, and J. Boyce, "Fast mode decision and motion estimation for jvt/h.264," in *Proc. Intl. Conference on Image Processing*, vol. 3, 14-17 Sept. 2003, pp. 853–856.

[105] P. Yin, A. Vetro, B. Liu, and H. Sun, "Drift compensation for reduced spatial resolution transcoding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 11, pp. 1009–1020, November 2002.

[106] P. Yin, M. Wu, and B. Liu, "Video transcoding by reducing spatial resolution," in *Proc. Intl. Conf. Image Processing (ICIP)*, 10-13 Sept. 2000, pp. 972–975.

[107] J. Zhai, K. Yu, J. Li, and S. Li, "A low complexity motion compensated frame interpolation method," in *Proc. Intl. Symposium on Circuits and Systems (ISCAS)*, 23-26 May, 2005, pp. 4927–4930.

[108] B. Zhang, "Generalized k-harmonic means-boosting in unsupervised learning," Hewlett-Packard Lab, Tech. Rep. HPL-2000-137, October 2000.

[109] Q. Zhang, Z. Ji, W. Zhu, and Y.-Q. Zhang, "Power-minimized bit allocation for video communication over wireless channels," *IEEE Trans. on Circuit and System for Video Technology, Special issue on wireless video*, vol. 12, no. 6, pp. 398–410, June, 2002.

[110] D. Zhong, "Segmentation, index and summarization of digital video content," Ph.D. dissertation, Graduate School of Arts and Sciences, Columbia University, 2001.

[111] X. Zhou, E. Li, and Y.-K. Chen, "Implementation of h.264 decoder on general-purpose processors with media instructions," in *Proc. of SPIE Visual Communications and Image Processing(VCIP)*, Jan. 2003.