NATIONAL UNIVERSITY OF SINGAPORE

# Video Encoder Optimization for Real-time Communication

by
Tan Yih Han

A thesis submitted in partial fulfillment for the
degree of Doctor of Philosophy

in the
Graduate School for Integrative Sciences and Engineering

January 2011

# *Abstract*

Effective video encoder control ensures that the performance of a video delivery system can be optimized under system limitation. The focus of this work is on algorithm design for encoder rate and complexity control. On top of being useful tools during video encoding, these algorithms also aid in the study of system performance under different constraints.

If available computational resource does not allow the entire encoding process to be carried out in time, a complexity scalable technique that ensures a graceful degradation of coding performance will be a valuable tool. Such a video encoding scheme is also useful in embedded video encoders of ubiquitous mobile devices which operate under energy restrictions. We design a video encoding scheme which allows the rate-distortion process to be carried out in a complexity scalable fashion.

Effective rate and power control are especially crucial for the design of power constraint real-time wireless video encoders. Power invested in the encoder can potentially reduce rate and improve picture quality while transmitter power determines the capacity of the communication channel. Hence, it is not obvious how power should be allocated between the source encoder and the transmitter, or if controlling the allocation can bring significant performance improvement. A complexity scalable encoding scheme that allows the control of complexity-coding performance trade-off enables the overall power optimization of a video encoding and delivery scheme. With an empirically derived complexity-rate-distortion model, both cross-layer resource allocation for individual user and resource allocation among multiple users can be studied.

# *Acknowledgements*

First, I wish to thank my advisors, Prof Lye Kin Mun, Dr Susanto Rahardja and the other researchers at the Institute for Infocomm Research whom I've worked with. Special thanks to Dr Li Zhengguo, Dr Tham Jo Yew, Dr Lee Wei Siong and Dr Yeo Chuo Hao who all share the misfortune of having to read stuff I've written. Also, Geraldine Ree, Lily Wong from I2R and Vivien Li from the NUS graduate school were always helpful and often tolerated my tardiness with administrative paperwork. I would also like to thank the thesis examiners, Dr Sun Huifang, Dr Bharadwaj Veeravalli and Dr Lin Weisi for their time and valuable comments. Last but not least, A*STAR Graduate Academy, for putting food on the table.

# Contents

# List of Figures

# List of Tables

# Abbreviations

| | |
|---|---|
| **MB** | **M**acro**b**lock |
| **AVC** | **A**dvance **V**ideo **C**oding |
| **SVC** | **S**calable **V**ideo **C**oding |
| **FGS** | **F**ine **G**ranularity **S**calability |
| **CGS** | **C**oarse **G**ranularity **S**calability |
| **MGS** | **M**edium **G**ranularity **S**calability |
| **DCT** | **D**iscrete **C**osine **T**ransform |
| **KLT** | **K**arhunen-**L**oeve **T**ransform |
| **GOP** | **G**roup **O**f **P**ictures |
| **HB** | **H**ierarchical-**B** |
| **MAD** | **M**ean **A**bsolute **D**ifference |
| **MSE** | **M**ean **S**quared **E**rror |
| **MV** | **M**otion **V**ector |
| **ME** | **M**otion **E**stimation |
| **MC** | **M**otion **C**ompensation |
| **MVF** | **M**otion **V**ector **F**ield |
| **RD** | **R**ate **D**istortion |
| **RDO** | **R**ate **D**istortion **O**ptimization |
| **PSNR** | **P**eak **S**ignal to **N**oise **R**atio |
| **SNR** | **S**ignal to **N**oise **R**atio |
| **Qp** | **Q**uantization **P**arameter |
| **NAL** | **N**etwork **A**bstraction **L**ayer |

# Chapter 1

# Introduction

## 1.1 Motivation

Rapid development in two areas of communication technologies has enabled the design of exciting visual communication applications: mobile communications and multimedia processing. Powerful signal processing techniques have significantly improved the coding performance of video coding schemes while advances in network architecture and increasing network capacity provide the channel required for the delivery of video data. These advances provide the framework on which mobile, wireless visual communication applications can be built.

Though modern video codecs may provide the compression capability required to maintain data rate below channel capacity, the processing power required to operate the source encoder, together with power needed to maintain the capacity of wireless channel, will drain batteries quickly even if the mobile computing platform can support encoding and transmission simultaneously. For this reason, reducing the complexity of video encoders while maintaining coding performance is a meaningful problem to look into. Ideally, fast decision algorithms should have minimal impact on the coding performance while reducing encoder complexity to meet system constraints. However, if the complexity of the encoder needs to be reduced further, a solution that can control the trade-off between encoder's complexity and coding performance will be a valuable tool.

Rate control is another aspect of video encoder control. Controlling the data rate, together with complexity control, can bring more flexibility to a rate and power constraint system, resulting in better performance. The control of bit-rate of encoded video data has become more challenging as video coding standards evolved. With header information taking up a significant portion of the encoded bit-stream, existing methods of rate control through modelling the rate-quantization behaviour of transformed coefficients may not be as effective.

With the techniques required to control complexity and rate of video encoders, it is interesting to study how such source encoders can improve performance of power-constrained systems. Since power that is used to improve the coding performance of the source encoder can also be channelled to maintain a larger channel capacity of the communication channel, a video encoding and transmission system that is capable of jointly optimizing encoded video bit-rate, source encoder power and transmitter power should bring better performance. With an empirically derived model, the effectiveness of cross-layer resource allocation for a single user and the performance of multi-user resource constrained system can be studied.

## 1.2 Research Scope

This thesis first develops algorithms (complexity reduction, complexity control and rate control) that are useful to both practical implementation of video encoders.

The next part of the thesis focuses on evaluating the value of a complexity (or power) scalable video encoder when optimizing system performance under constraints. First, cross-layer power allocation between the source encoder and a wireless transmitter is studied. Next, the distributed optimization of a network of wireless encoders is analyzed.

## 1.3 Main Contribution

- Through the analysis of the rate-distortion performance of different layers of a scalable video, designed fast rate-distortion algorithms suitable for multi-layer encoder optimization [1].
- Designed a new framework for fast motion estimation suitable for the coding of hierarchical-B pictures [2].
- Designed a new model-based video rate control scheme that allows fast adaptation and accurate rate control[3][4].
- Designed a singularly-parameterized complexity scalable video encoding scheme that allows complexity-coding performance trade-off during encoding [5][6] and cross-layer resource allocation in layered video encoding[7].
- Modelled the complexity-rate-distortion behaviour of aforementioned complexity scalable encoder. The model allows the use of the encoder for overall system optimization[8].
- Used the proposed complexity scalable and rate control algorithms to study the possibility of joint source coding-power control for video transmission under power constraint[9][10].
- Studied the implication of complexity scalable during distributed optimization of wireless video encoders in a wireless network[11][12].

## 1.4 Video Coding Standards

The need to compress video data, either for transmission or storage is obvious. Despite increasing communication capacity and storage space, the demand for better compression techniques has not diminished as video resolution requirement grows. Data compression includes the removal of redundancy from a signal to allow more compact representation. It is clear that the predictability of a signal is intimately related to the idea of redundancy. Therefore, it is not surprising that the data rate of transmitted video can be drastically reduced with predictive coding.

A video coding standard is the language though which a video encoder and decoder communicate. A video standardization process defines the syntaxes and semantics of a compliant video bit-stream, ensuring that all compliant decoders can understand an encoded file produced by any compliant encoder.

Although the syntaxes and semantics of the language are finalized with the standardization of a video codec, there remains a large degree of flexibility when designing and implementing a compliant encoder.

### 1.4.1 Earlier Codecs

From the dozens of pages required to describe the earlier coding standards such as H.261 and MPEG1, modern coding standards now require hundreds of pages to describe. Despite the addition of numerous tools that led to the improvement of modern codecs over legacy ones, the core technology behind the codecs remains similar. Motion compensated prediction to remove inter-frame redundancy followed by transform coding for energy compaction that allows effective quantization has been proven to be an exceptionally effective method to compress video data.

Video compression typically operates on square-shaped groups of neighboring pixels, often called macroblocks. These pixel groups or blocks of pixels are compared from one frame to the next and the video compression codec (encode/decode scheme) sends only the differences within those blocks.

The success of hybrid video encoders can be attributed to several key components:

- Motion compensated predictive coding that reduces redundancies that exist between temporally distinct video frames. Predictive coding reduces the power of the symbols that are coded; fewer bits are required to code prediction residues compared to actual pixel values.
- Transform coding of prediction residues leads to coefficients that are less correlated than the original samples; adjacent samples typically have substantial correlation and

separate quantization of each would be an inefficient way to encode them. In addition, transformation leads to energy compaction such that information is concentrated in only a few transform coefficients. As a result, they can be quantized and encoded more efficiently.

- Entropy coding of quantized transform coefficient reduces the number of bits required to represent them. The goal of entropy coding is to reduce the average number of symbols sent with no loss of fidelity in the decoded signal.

## 1.5 H.264/AVC

For a number of years, work on the "Advanced Video Coding" standard, now known as ITU-T Recommendation H.264 and as ISO/IEC 14496 (MPEG-4) Part 10 occupied the video coding community. Even after the standardization work is done and coding standard specified, the H.264 video coding scheme continues to be an active area of research. Due to the excellent coding performance of the codec, both the reseach community and industry players expect it to be widely adopted.

H.264 has many new features that improve picture quality and compression performance [13]:

- H.264 uses quarter-pixel precision for motion compensated prediction.
- In-loop deblocking that removes artifacts caused by block-based prediction and transform improves both perceptual quality of reconstructed pictures and coding performance [14].
- H.264 allows the use of smaller block sizes during motion compensation. The use of finer partition during the motion compensated prediction improves the prediction accuracy and reduces bits required to code the residue. This can potentially lead to better coding performance [15].
- Weighted prediction allows prediction signal to be weighted. This significantly improves the coding of video frames containing fades [16].
- Improved SKIP mode allows motion vectors to be derived from neighbouring block despite expending no bits for coding motion vectors [17].
- Spatial prediction for intra frames reduces the variance of data that has to be coded for intra coding. Edge directed prediction removes redundancy within a picture effectively, resulting in fewer bits needed for coding the picture.
- The use of arithmetic coding, adaptive codes and context modelling [18].

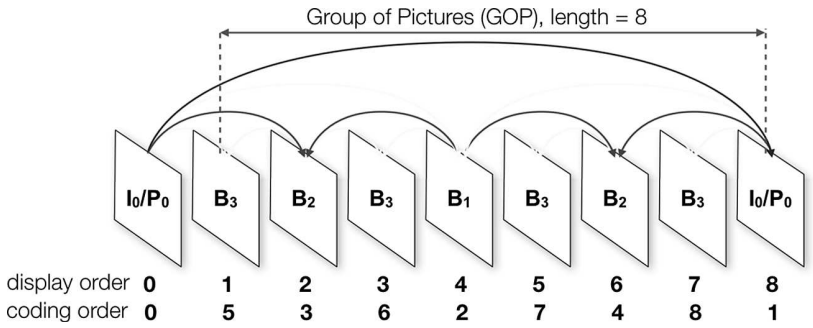



FIGURE 1.1: A hierarchal B-pictures coding structure with 4 temporal levels, indicated by the subscript of picture type (I, P or B).

- A DCT-like integer transform that resulted in an exact definition of the inverse transform [19].
- Multiple reference frames can be used for motion-compensated prediction.

## 1.6 Scalable Video Coding

Through scalable video coding [20][21], an encoded video file can be decoded to fulfill several frame-rate, resolution or quality requirements after appropriate extraction. To achieve this, each video is encoded into layers. Enhancement layers successively refine their base layers, resulting in decoded videos that are of higher resolution, higher frame rate or better (PSNR) quality. The main challenge of scalable video coding is to remove inter-layer redundancy. Ideally, the coding performance of scalable video should approach that of a single layer codec when producing decoded pictures of the same quality. This is usually not the case. Therefore, the flexibility of being able to represent videos of different resolution, frame rate and quality comes at a cost of a coding performance penalty.

As consumers begin to access video files on a wider variety of platform, service providers are increasingly tasked with supplying optimized video streams to devices as disparate as cell phones, personal computers and set-top boxes. Scalable video coding, with its ability to fulfill different requirements with a single file brings convenience and flexibilty that service providers may find appealing.

### 1.6.1 Hierarchical-B coding Structure for Temporal Scalability

When temporal scalability is a requirement in coded sequence, the video encoder has to avoid using frames of higher temporal levels for prediction when encoding a particular frame. This can be achieved by encoding each Group of Pictures as a hierarchy of frames (Fig. 1.1). In a temporal scalable video, the video frames in the lowest temporal level form the base layer

and each subsequent temporal level becomes an enhancement layer that enhances the frame rate of the video.

#### 1.6.1.1 Bi-directional Predictive Pictures

Bi-directional pictures are pictures that are coded using pictures in both temporal directions as reference frames. Blocks in a bi-directional frames have the choice of using either past frames or future frames (or a combination of both) during motion compensated prediction. A newly uncovered object in the current coded frame cannot be successfully predicted with past frames; this will inevitably affect prediction and coding performance. This problem can be partially alleviated with forward prediction or bi-prediction. Hence, the inclusion of bi-directional predictive pictures usually brings coding performance improvement. When encoding for low-delay applications, temporal scalability can also be achieved with hierarchically-arranged P-frames.

During the encoding of bi-directional predictive frames, the reference frames and coded frames are usually further apart temporally. This poses a challenge for available fast motion estimation algorithms as most of them reduced the number of search points by assuming that a good match for the coded block can be found close to the co-located block in the reference frame. In Chapter 2, this weakness of available fast search techniques will be highlighted and a new motion estimation framework that overcomes the difficulty will be introduced.

### 1.6.2 Fine Granularity Scalability

In layered scalable encoding schemes, the video is encoded into a base layer and several enhancement layers and every layer contributes towards the quality of the video. However, an entire layer has to be received by the decoder to enhance the video quality, so the problem of poor channel adaptivity remains.

Similar to layered video coding, a fine-grained scalable video is encoded into several layers. The enhancement bit stream of a FGS layer, however, can be truncated at any point and still enhances the quality of a video. In MPEG-4 Part 2 (Visual), FGS is achieved through bit-plane coding of the transform coefficients [22][23]. Instead of coding quantized transform coefficients through run-level coding as decimal integers, each coefficient is represented by its binary form in the FGS scheme. For a transformed block, a bit plane is formed by taking one bit from each value of transform coefficients at the same significant position. Now the improvement in quality brought about by the enhancement layer is proportional to the number of bits received by the decoder.

The value of fine-grained scalable encoding is accentuated when a group of clients of different bit-rates are requesting for the same video. In a video-on-demand system, a single file in the media server can fulfill a range of bit-rate requirement.

For video communication, applications that encode for the requirement of a single client can be designed with real-time quantizer adjustments. This technique is, however, inadequate when a single source is serving clients of different bit-rate requirements. In this situation, fine-grained scalable video encoding provides a convenient alternative.

The effective encoding of layered video requires the optimization of each layers. This inevitably increases the complexity of the encoder. When faced with the requirement to design a low power encoder and a coding standard that provides a large number of possible coding options that may increase the power consumption of video encoding, the ideal solution would be one that reduces complexity with negligible effect on coding performance. The aim of achieving low complexity encoding has motivated many work in literature. We use this approach in Chapter 2 where fast decision algorithms suitable for the encoding of layered videos and a fast motion estimation method suitable for a hierarchically arranged video frames will be described.

Of course, the encoder power consumption or complexity can only be reduced to a certain extent without hurting coding performance. If further complexity reduction is required, a method that allows graceful degradation of performance as complexity is reduced would be useful. We described such a scheme in Chapter 3.

## 1.7 Rate-Distortion Optimization in Video Coding

Shannon's celebrated work [24] marks the beginning of classical rate-distortion theory. Rate-distortion theory is concerned with the optimal representation (with fewest number of bits) of source data given certain fidelity requirement. This ability to trade-off rate and fidelity enables lossy compression: when the rate constraint does not allow a lossless description of the source data, the number of bits used to describe the source data can be reduced at a cost of reduced decoded signal fidelity.

Rate-Distortion optimization during video encoding answers the following question for each unit of video data encoded: out of the variety of encoding techniques provided by the video coding standard, what is the best way to represent the video data? To produce a video bit-stream that is compliant to a particular video coding standard, it is required that the bit-stream must be decodable by a standard compliant decoder. That is, the video encoder and decoder must be speaking the same language. There is, however, no specification on the encoding process. In another words, two different encoder implementations, producing bit-streams compliant to the same video coding standard, can differ greatly in coding performance.

The rate-distortion-optimized encoding strategy does not only improve the coding performance in relation to older techniques, but also allows a fair comparison between different hybrid video coding standards in terms of coding efficiency.

To make a rate-distortion optimized decision during the encoding of a block of video data, the encoder has to compute the rate-distortion cost associated with each coding mode before eventually choosing the best (lowest cost) mode that will be used to code the block. For the decision to be truly optimized, the right combinations of all the options for the various coding tools have to be chosen. Hence, a block has to be encoded numerous times before the encoder can arrive at the best set of parameters for it. This increases the computational complexity of the encoder. While an offline encoder can be tolerant of encoder complexity, real-time applications require that encoding be done within time constraints. Therefore, it is worthwhile to design an encoding scheme that allows the complexity-coding performance trade-off of a video encoder. The proposed complexity scalable scheme is described in Chapter 3.

Complexity scalability, where the computational complexity of an encoder can be scaled with a trade-off in coding performance, is a useful tool. When computational resource is limited but a fast implementation of the encoder is required, the complexity of the encoder can be scaled down to ensure that encoding can be done on time. Real-time encoding is required for applications such as live broadcast, surveillance or video communication. Since these applications may be built on a wide variety of computing platforms, to make full use of computational resource while ensuring that encoding completes on time will be difficult without an effective complexity scalable solution.

One obvious advantage of a power scalable is the possibility to design power adaptive application where the complexity of the encoder is adjusted according to varying system resource. It is also useful for implementing a video encoder that has to be deployed on different platforms; the power scalable nature of the encoder will ensure that it operates within system constraints despite varying resource level across different computing platforms. The design of such an encoder is also discussed in Chapter 3.

## 1.8 Video Rate Control

Controlling the rate of encoded videos is important for several reasons. If there is limited memory buffer at the encoder, rate control of the encoded video will reduce buffer overflow. Rate control also ensures that the bit-rate of video stream remains within the capacity of transmission channel. The challenge of designer an encoder that does both rate-distortion optimization and rate control is highlighted in Chapter 4. In the chapter, a new rate control scheme that enables tight video rate control is also described.

With models that track the variation of both the header and texture bits with quantization parameter, we resolve the RDO/rate control conundrum by considering $Q_p$ for quantizing

transform coefficients and the $\lambda$-determining $Q_p$ as separate parameters that are obtained from a constrained optimization procedure. Two one-step model update processes help the rate control scheme track the changing characteristics after the encoding of each basic unit. Since the updates only require information from the previously encoded basic unit, they provide fast approximations for the proposed rate control scheme to make $Q_p$ adjustments. In contrast, many available rate control schemes require encoding information from several basic units to carry out regression-based model updates. These schemes often suffer from slow model adaptation and thus do not provide effective rate control for sequences with fast varying characteristics.

Encoder rate control (Chapter 4) together with complexity control (Chapter 3) are important tools for the design of live video encoding and delivery system.

## 1.9 A Video Encoding and Transmission System

There is a wide range of applications that require video encoding and streaming over wireless communication networks. The increasing computing power of mobile devices, coupled with the expanding bandwidth of wireless networks means that we may soon experience ubiquitous live streaming from mobile devices. When videos are encoded and streamed from devices with limited power or energy, judicious use of power supply and calculated allocation between transmitter and video encoder should improve system performance.

For a single user, power allocated to the encoder can potentially improve coding performance and reduce the number of bits required to code the video sequence. However, if the system is operating under power constraint, investing in the source encoder channels power away from the wireless transmitter. This will reduce the capacity of the wireless channel since capacity. Given the system parameters that the system are operating under, there should be an optimal allocation between the source encoder and the wireless transmitter. However, it is not clear how much improvement to the system will such cross-layer optimization brings.

The complexity (power consumption) of the video encoder described in Chapter 3 can be easily controlled with a single parameter. That is, the complexity-performance trade-off of the video encoder can be controlled with a single parameter. Through experiments, an empirically-derived model of the encoder is obtained. With the model, simulation is carried out to gain insights into the effects of encoder-transmitter power allocation on system performance (Chapter 5).

When several video encoder-transmitter units are sharing a wireless network, being able to adjust power allocation at an individual level can also affect the performance of the entire system. In a CDMA-like network, each user spreads its signal over the entire bandwidth such that for any particular user, other users' signals appear as pseudo white noise. Therefore, when a user increases his transmitter power, improvement to his signal-to-noise ratio comes

at the cost of increase noise for the other users sharing the network. In Chapter 6, simulation is carried out to study the effect of encoder power control and its possible effects on the network of users. We consider the scenario where each wireless user encodes its video stream and competes with other users for the wireless medium to send the video data to the base station. When in operation, each user will have control over its encoder parameters and transmitter power. Our analysis on distributed power control is useful for the design of a network of wireless video encoders.

## 1.10 Organization

The rate distortion process of layered videos is studied in Chapter 2 and algorithms useful for reducing the complexity of SNR scalable video are designed. A framework useful for incorporating existing fast search algorithms into motion estimation for Hierarchical-B frames is also described.

Chapter 3 describes a complexity scalable encoding scheme that allows a coding performance-complexity trade-off in video encoders while Chapter 4 describes a new rate control scheme.

In Chapter 5, the rate-distortion-complexity behaviour of the complexity scalable encoder is analyzed before simulation results that show the value of both rate and complexity control of an encoder is presented.

With the empirical models derived in Chapter 5, the behaviour of wireless encoders sharing a wireless network is studied in Chapter 6. The effect of encoder-transmitter power allocation on the performance of all participating users is also studied.

# Chapter 2

# Fast Encoder Optimization

## 2.1 Introduction

In this chapter, the rate-distortion optimization process when encoding layered videos and the motion estimation across hierarchical-B pictures with a GOP are studied.

When faced with the requirement to design a low power encoder and a coding standard that provides a large number of possible coding options that may increase the power consumption of video encoding, the ideal solution would be one that reduces complexity with negligible effect on coding performance. The aim of achieving low complexity encoding has motivated many work in literature. We use this approach in this chapter where fast decision algorithms suitable for the encoding of layered videos and a fast motion estimation method suitable for a hierarchically arranged video frames will be described.

First, we introduce a low complexity fine granularity scalable video encoder that refines both residue and motion information in the quality layers. The current Scalable Video Coding draft shows that significant gains can be achieved when each enhancement layer undergoes the motion estimation/motion compensation process with its own motion vector field. However, given the high computational cost of ME/MC, a motion refined FGS scheme can be expensive to implement. The proposed scheme controls the MB mode allowed in the base layer and channels computational resource to refine motion in enhancement layers.

Through a proper selection of Lagrangian multiplier for the generation of the first MVF, it is possible to design a low complexity FGS encoder that has good overall coding performance. A simplified motion refinement scheme is also adopted for selected MBs in enhancement layers by exploiting the correlation of MB type information between successive layers to further reduce the complexity of FGS encoder. Meanwhile, a framework of rate-distortion-complexity optimization is proposed for the FGS by considering the interpolation complexity during the

ME/MC in each layer. The FGS decoder can be simplified through the reduction in the number of interpolations.

In the second part of this chapter, a framework that integrates with existing fast ME methods and improves their motion prediction accuracy when employed in the HB structure (or any coding structures where there exists intermediate frames between current and reference frames) by extending their effective motion search range through successive motion vector interpolation along the MB's motion trajectories across the frames within the GOP is proposed.

## 2.2 Rate-Distortion Optimization in Video Encoding

Rate-distortion optimization ensures optimal rate allocation among every coding unit in the coded sequence. For each coding unit, the best coding mode is chosen as the the one that minimizes the Lagrangian cost: $J = D + \lambda R$. The non-negative Lagrange multiplier, $\lambda$ allows the selection of a specific trade-off point. By considering operating points at constant slope ($\lambda$), all coding units are operating at the same marginal return for any extra bits invested. Hence bits cannot be taken away from one coding unit and given to another to result in an overall quality improvement.

This makes video encoding inherently computationally demanding. All combinations of coding options have to be tested to compute each of their rate-distortion performance. One obvious way to reduce the complexity of video encoding is to leave out a subset of coding modes that are judged to be less probable to be picked. Such a strategy may lead to sub-optimal performance if modes that are omitted include the optimal mode.

## 2.3 Fine Granularity Scalability

Scalable video encoding produces bitstreams that can fulfill different spatial, temporal and SNR requirements through appropriate extraction. Spatial scalability is achieved by encoding a video into layers of different resolutions. Various inter-layer prediction techniques are designed to reduce inter-layer redundancy and improve coding efficiency. Temporal scalability is achieved through a hierarchical prediction structure where frames from higher temporal levels compared to the encoded frame are not used for reference. This chapter focuses on SNR scalability which allows a coded bitstream to adapt to different channel capacity. The network channel capacity can vary over a wide range depending on the type of connection; it can also be unpredictable due to network traffic. To overcome these problems, it is no longer sufficient for a video encoder to optimize the video quality at a given bit-rate; video quality must now be optimized over a range of bit-rates [25].

Layered scalable coding techniques were first introduced to provide the SNR scalability. In such encoding schemes, the video is encoded into a base layer and several enhancement layers and every layer contributes towards the quality of the video. However, an entire layer has to be received by the decoder to enhance the video quality, so the problem of poor channel adaptivity remains. Fine granular scalability (FGS) provides a better solution. Similar to layered video coding, a video is encoded into several layers. The enhancement bit stream of a FGS layer, however, can be truncated at any point and still enhances the quality of a video. In MPEG-4 Part 2 (Visual), FGS is achieved through bit-plane coding of the transform coefficients [22][23]. Instead of coding quantized transform coefficients through run-level coding as decimal integers, each coefficient is represented by its binary form in the FGS scheme. For a transformed block, a bit plane is formed by taking one bit from each value of transform coefficients at the same significant position. Now the improvement in quality brought about by the enhancement layer is proportional to the number of bits received by the decoder.

FGS has been well studied in the past decade. Only the base layer pictures are involved in the temporal prediction in the conventional FGS scheme [23]. Since the reference pictures are of poor quality, the prediction performance and consequently the coding efficiency are also relatively poor at high bit rate. In [26] and [27], progressive FGS frameworks that use reference pictures with increasing quality were introduced. The work in [28] described a scheme that estimates a set of motion information from each reference and then uses the set of motion information that gives the best rate distortion performance across the layers. Huang et al. [29] proposed a scheme that substantially improves coding efficiency and at the same time provides a good trade-off between efficiency and error drift through controlling the number of bitplanes used and the amount of prediction leak during the construction of enhancement layer reference frames.

During FGS coding in the joint scalable video model (JSVM) [30], the encoder can choose to close the prediction loop at the highest quality point or at the lowest and highest quality points. Fine grain SNR scalability is achieved by encoding successive refinements of the transform coefficients by repeatedly decreasing the quantization step size. The coefficient coding is done in two passes, a significant pass and a refinement pass. A significant pass encodes coefficients that have a value of zero in the previous layers. A refinement pass refines the precision of non-zero coefficients in the previous layer. To extract a requested bit-rate from the overall bit-stream, the progressive refinement network abstraction layer (NAL) units are truncated at appropriate points. FGS has recently been removed from the SVC draft. SVC now supports medium-grain scalability (MGS), which is conceptually similar to coarse-grained scalability (CGS). But with MGS, the adaptivity is significantly increased, since each enhancement layer packet can be discarded and packets which, when absent, have the least impact on coding efficiency are discarded first.

Besides the poor quality of reference pictures, another problem for FGS coding is the poor trade-off between motion and residue information at high bit rate [31]. The MVF is usually generated at low bit rate [23], the same MVF is used for all FGS layers and only the residual information is predicted and refined from the previous FGS layer. Recently, Winken et al. [32] applied adaptive motion refinement in FGS slices, and found that the PSNR gap between a FGS video coding scheme and a single layer coding scheme can be reduced if enhancement layers also undergo motion estimation and acquire a different set of information. This scheme is attractive as it achieves significant coding gain without increasing decoder complexity. This scheme is however, computationally expensive at the encoder end. Another disadvantage of the current JSVM encoder control is that the coding efficiency losses against single-layer video coding are unequally distributed in different layers. While the coding efficiency of the base layer is almost the same as single-layer coding, there is usually a significant loss in coding efficiency for the enhancement layers. It is difficult to meet the requirement of a 10% gap between the SVC and single-layer video coding in all layers by using the current JSVM encoder control. In a practical application of SVC where customers may be more interested in an enhancement layer, techniques that allow an encoder to improve enhancement layer coding efficiency at the expense of base layer performance may be necessary[33].

We shall propose several fast decision algorithms to simplify the motion refinement process during FGS encoding while maintaining the coding efficiency. The impact on the coding efficiency of enhancement layers is taken into consideration when fast mode decision algorithms are designed in the base layer. This is achieved by deriving the Lagrangian multiplier used during rate distortion optimization with quantization parameter (Qp) that is smaller than Qp for the quantization of transform coefficients. That is, the Lagrangian multiplier is decoupled from the coefficient quantization parameter. This is different from existing fast mode decision schemes for single-layer video coding where only the coding efficiency of the single layer is considered [34]. To further reduce the overall complexity, motion information is only refined for selected MBs instead of every MB in enhancement layers. Since the partition of an MB in an enhancement layer is usually finer than the corresponding MB in its base layer, partition of an MB in an enhancement layer is restricted according to the corresponding MB in its base layer. The proposed scheme also controls the MB mode allowed in the base layer and channels computational resource to refine motion in enhancement layers. With the proposed methods, the complexity of motion refined FGS is significantly reduced. Furthermore, the gaps between the SVC and the single-layer video coding can be adjusted in all layers. This feature can be used to ensure that performance gap from a single layer codec are near uniform across all layers.

In section 2.4, the necessity of motion refinement for the FGS is studied and the motion refined FGS scheme in SVC is described. A simplified motion refined FGS encoder is proposed in Sections 2.5 and 2.6. By controlling the MB type in the base layer and then refining the motion information selective MBs in an enhancement layer according to its base layer MB type, the number of rate distortion computation can be dramatically cut down with little impact on

coding efficiency. A FGS decoder complexity reduction method is described in Section 2.7 where we explain how the number of interpolations for sub-pixel motion compensation at the decoder can be reduced during the motion refinement process. Experimental results will be presented in Section 2.11 before we conclude in Section 2.12.

## 2.4 Motion Refined FGS Scheme

It is suboptimal to use the MVF that is generated in the base layer when coding enhancement layers. Since only residue information is refined in such schemes, the coding performance is poor at high bit rate due to the poor tradeoff between motion and residue information. There exists a gap of 1 - 3dB when such schemes are compared to single layer coding schemes at high bit rates. Motion information refinement in the enhancement layers can lead to better prediction result. However, does the improvement in quality brought about by the inclusion of another set of motion information justify the extra bits required to code the motion information?

Adaptive motion refinement for FGS slices was introduced in [32] which removes the restriction of using only one set of motion information throughout the SNR layers. If the encoder decides that coding performance gains do not justify the inclusion of a different set of motion information, no such information will be coded for an MB in the enhancement layer and a base layer skip flag signals to the decoder that the information of the base layer MB is used and the decoding of the FGS MB continues with the decoding of the transform coefficients. During decoding, before the first transform coefficient of a non-intra coded MB is read, the decoder will read a base layer skip flag which signals if the motion information (partition type and motion vectors) of the co-located base layer MB is used. If the flag signals the presence of a new set of motion information, the decoder will proceed to read them in. Since the options of motion vector prediction and residue prediction are open to a motion refined FGS layer MB (similar to a CGS enhancement layer MB), the appropriate motion prediction and residue prediction flags have to be read by the decoder and the intended predictions have to be carried out. Refinement of motion information in progressive refinement slices, similar to that in CGS schemes was found to be justifiable as PSNR gain of 1dB is possible when the bit rate is low in the base layer. As the base layer bit-rate increases, the incentive to carry out motion refinement in the FGS enhancement layers decreases as the problem of suboptimal motion vector field is no longer as acute a problem. *Here, it is assumed that the base layer is at low bit rate.* It is interesting to note that there is no complexity increase at the decoder since motion-compensated prediction is done only once in an enhancement layer. However, the complexity of encoder is increased significantly. It is desirable to provide effective algorithms to reduce the complexity of encoder with negligible impact on the coding efficiency.

## 2.5 Mode Decisions in The Base Layer

In a typical H.264/AVC encoder [35] [36], inter frame motion estimation is performed by computing rate-distortion (RD) cost for all possible MB modes: 16x16, 16x8, 8x16, 8x8, 8x4, 4x8 and 4x4 (Fig. 2.1). The mode that results in the least RD cost is chosen. Such exhaustive searching will provide the optimal rate distortion performance but is also computationally expensive. The high computational complexity of the full search algorithm has motivated a host of suboptimal but faster search strategies [34]. Although there are many interesting fast mode decision schemes for H.264, they only consider the coding efficiency of individual layers; the multiple layer nature of scalable videos may require a cross layer approach. There are two objectives when fast mode decision algorithms are designed in the base layer. One is to reduce the complexity of base layer with little impact on its coding efficiency and the other is to obtain a good tradeoff among coding efficiency in all layers.

The proposed scheme deprives the base layer of computational resource by restricting the partition type that each MB can assume. The candidate set is {skip, 16x16, 16x8, 8x16}. Although fine partitions may be discriminated against during rate-distortion optimization anyway, the extra rate distortion computation at the encoder is unavoidable. By controlling the MB type in the base layer and then selectively refining the motion of enhancement layer MB according to the base layer MB type, the number of rate distortion computation can be dramatically cut down with little impact on coding efficiency. This encoder optimization technique is especially effective when encoding a video with a low bitrate base layer and a large range of extractable bitrates.

The second objective can be achieved by optimizing the motion information for a bitrate higher than the bitrate of the base layer when the MVF is generated in the base layer. This can be achieved by using a different quantization parameter to compute the Lagrangian multipliers during the acquisition of motion information. The encoder improves coding performance in enhancement layer while compromising the coding efficiency in the base layer, possibly resulting in a near uniform performance gap between the SVC and single-layer video coding in all layers. It is required that such a gap is about 10% at all layers by the on-going SVC standard. Moreover, an implementation of the FGS scheme that cannot afford the encoder complexity that comes with motion refinement in the FGS layers, may find compromising the coding efficiency of the base layer and achieving better performance in the FGS enhancement layers an attractive option.

Suppose that $Qp_{base}$ is the quantization parameter used for coding transformed coefficients of the base layer MB and $\alpha$ is an auxiliary variable that will determine the bitrate the motion information that is acquired is optimized for. The proposed fast mode decision scheme in the base layer is formulated as below.

During motion estimation, the base layer will select the motion vector, $\tilde{MV} = (mv_x, mv_y)$ that minimizes the following cost function:

$$J(\tilde{MV}, \lambda_{\texttt{motion}}) = \texttt{SAD} + \lambda_{\texttt{motion}} R(\tilde{MV}) \tag{2.1}$$

where SAD is the sum of absolute difference between original signal and the predicted signal, $R(\tilde{MV})$ is the number of bits required to code the motion vectors, and

$$\lambda_{\texttt{motion}} = 0.92 * 2^{(Qp_{base} - \alpha - 12)/6}. \tag{2.2}$$

After motion estimation for each mode, the mode that leads to the lowest rate-distortion cost is selected. This is usually done through minimizing the following cost function:

$$J(\texttt{mode}, \lambda_{\texttt{mode}}) = \texttt{SSD} + \lambda_{\texttt{mode}} R(\texttt{mode}), \tag{2.3}$$

where `mode` is the MB mode that is chosen from the set {skip, 16x16, 16x8, 8x16}, SSD is the Sum of Squared Difference, $R(\texttt{mode})$ is the number of bits needed to code the MB and

$$\lambda_{\texttt{mode}} = \lambda^2_{\texttt{motion}} = 0.85 * 2^{(Qp_{base} - \alpha - 12)/3}. \tag{2.4}$$

In this manner, the Lagrangian multiplier used during rate distortion optimization of the base layer is no longer only dependent on the quantization parameter used in the base layer. This technique has the effect of biasing the optimality of the acquired motion information towards certain bit-rate point that can be higher than what is achievable with the base layer.

Although it is possible to improve enhancement layer coding efficiency by controlling the bitrate at which the motion information is optimized for, overall coding performance may still suffer if the FGS layers span a wide range of bitrates. To ensure good coding efficiency throughout the entire range of bitrates, we carry out motion refinement for selected MBs in FGS enhancement layers.

With the overall coding performance in mind, we seek to achieve good motion and residue information trade-off through these three methods:

- Decoupling the Qp for motion estimation/motion compensation (ME/MC) and the Qp for the coding of coefficients. This provides a layer of relatively low bitrate the option of carrying a set of motion information that is optimized for a higher bitrate.
- Channeling part of the computational resource used for the ME/MC in the base layer to enhancement layers.
- Low cost refinement of motion information that will be described in Section 2.6.1 for selected MBs in enhancement layers.

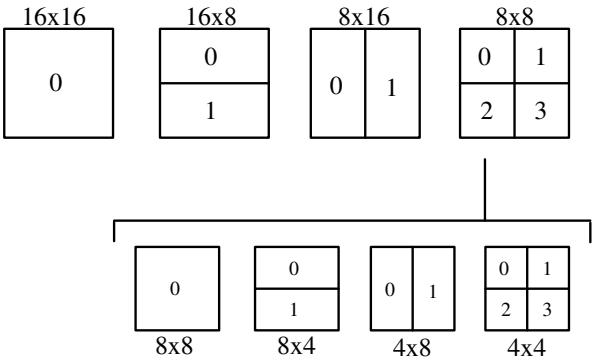



FIGURE 2.1: Possible partitions of a 16x16 MB

| | Foreman | | City | |
|---|---|---|---|---|
| | Base | Enhance | Base | Enhance |
| SKIP | 34060 | 26186 | 31908 | 19804 |
| 16x16 | 17247 | 23628 | 21572 | 32204 |
| 16x8 | 2055 | 2653 | 710 | 1030 |
| 8x16 | 1574 | 2203 | 1197 | 2120 |
| 8x8 | 251 | 517 | 45 | 274 |
| total | 55187 | 55187 | 55432 | 55432 |

TABLE 2.1: Number of each MB type

## 2.6 Simplified Motion Refinement Scheme in Enhancement Layers

### 2.6.1 Mode Constraint In Enhancement Layer

In this section, we explore the possibility of reducing the set of candidate MB modes used in the enhancement layers by making use of base layer MB mode information.

MV prediction exists in the base layer to exploit the correlation between neighbouring MVs. The MV of a partition can be predictively coded against that of neighbouring partitions that have already been coded. In an enhancement layer, the encoder has the extra option of predicting the MV from the corresponding partition in the base layer [37]. In a typical encoder, all MB modes are tested with and without residue prediction and MV prediction to determine the set of motion information that gives the best rate distortion performance. Therefore, any motion estimation for an enhancement layer MB is also going to be time consuming. The wide range of coding options available to an enhancement can potentially improve coding efficiency but inevitably increases the computational cost.

To achieve better image quality, the Qp in the enhancement layer is lower than that in the base layer. In this way, transformed coefficients in the enhancement layers are quantized less heavily, leading to better fidelity. When better prediction performance is required, partitions can be as small as 4 pixels by 4 pixels, if deemed justifiable after rate distortion computation.

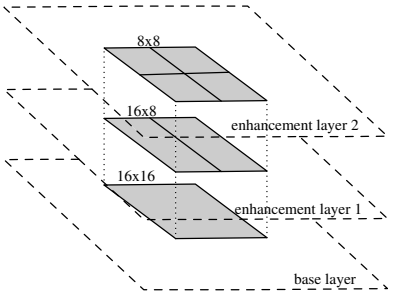



FIGURE 2.2: Hierarchical partitioning of MB across SNR layers

| Base Layer MB mode | Enhancement layer MB mode |
|---|---|
| SKIP | 16x16 |
| 16x16 | 16x8, 8x16 |
| 16x8 | 8x8 |
| 8x16 | 8x8 |
| 8x8 | no motion refine |

TABLE 2.2: Possible enhancement layer refinement MB mode given base MB mode

During mode selection, a Qp dependent $\lambda$ controls the trade-off between rate and distortion. Its value is generally lower at high bit rates when better image quality is desirable. Smaller partitions are more likely in an enhancement layer.

We encoded two sequences into two layers (base and FGS enhancement layer with motion refinement) using Qp 46 and 40 respectively and recorded the number of each MB type in each layer in Table 2.1. It is obvious that the encoded picture is more finely partitioned in the enhancement layer after ME/MC. Through experiments, we also found that majority of MBs in the enhancement layer in [32] do not undergo refinement. Our experiments have also revealed that if the rate-distortion computation determines that it is justifiable for an enhancement MB to carry its own motion information, it is highly likely that the base and enhancement MBs will be of different types. With this information, it is already possible to design a faster encoder by leaving out the base MB type during the motion refinement in enhancement layers.

The proposed encoding scheme restricts the MB type of an enhancement layer MB according to its base layer MB type (Table 2.2); the rate distortion optimization process for an enhancement layer MB is only carried out on the MB types that partitions the corresponding area more finely than the base MB type. It is carried out by computing rate distortion cost:

$$J(\texttt{mode}', \lambda_{enh_{\texttt{mode}}}) = \texttt{SSD} + \lambda_{enh_{\texttt{mode}}} R(\texttt{mode}'), \tag{2.5}$$

where $\texttt{mode}'$ is the MB mode that is being forced on the enhancement MB according to the base MB type (Table 2.2), and the value of $\lambda_{enh_{\texttt{mode}}}$ is $0.85 * 2^{(Qp_{enhance} - \alpha' - 12)/3}$ with $Qp_{enhance}$ be the Qp for the quantization of residual information in the corresponding enhancement layer. If the cost is smaller than that when the enhancement layer MB uses base layer motion

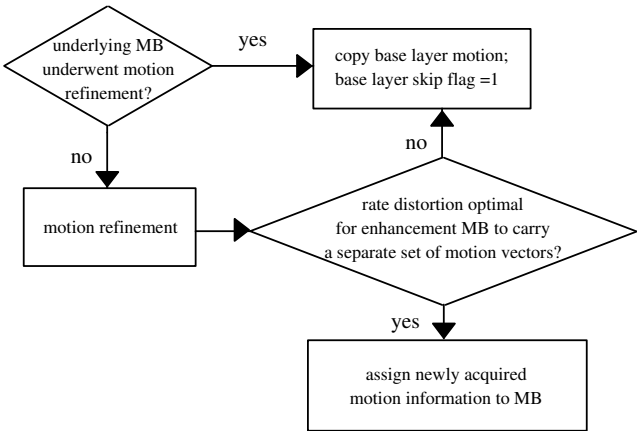



FIGURE 2.3: Avoiding motion refinement in co-located MB in successive layers; encoder decision in second and third FGS layers

information, a different set of motion information is coded for the enhancement layer MB. The rate distortion decision the encoder has to make is: should the motion data for the enhancement layer MB be refined by using the next finer MB type or should the MB use the base layer motion information?

The result is a hierarchical partitioning of any 16x16 area across SNR layers (Fig. 2.2). By controlling the MB type used, an area in the enhancement layer will be more finely partitioned than the corresponding area in the base layer. With this restriction, the number of rate distortion computations decreases dramatically. Our experiment shows that substantial gain from the motion refined FGS can still be achieved despite the simplified ME/MC process.

### 2.6.2 Motion Refinement for Co-located MB in Successive FGS Layers

To further reduce the number of rate distortion computations required in the FGS enhancement layers, motion information is not refined in an MB in enhancement layer when the co-located MB in the underlying FGS layer has undergone motion refinement. This decision process takes place in the second and third enhancement layers. We found that this has very little impact on the gain in the enhancement layers after motion refinement. Fig. 2.3 shows the encoder decision making process.

## 2.7 Rate-Distortion-Complexity Optimized Motion Refinement

The complexity of encoder has been studied in the previous two sections. The complexity of decoder will be addressed here by using a framework of rate-distortion-complexity optimization (RDCO).

Ugur et al. [38] proposed an encoder that biases easy-to-decode motion vectors; motion vectors that require more interpolation to decode are discriminated against. This method is extended to reduce the complexity of the FGS decoder.

To discriminate against motion vectors that are complex to decode [38], the cost function in (2.1) is modified as

$$J(\tilde{MV}, \lambda_{\texttt{motion}}, \lambda_c) = \texttt{SAD} + \lambda_{\texttt{motion}} R(\tilde{MV}) + \lambda_c C(m) \tag{2.6}$$

where $C(m)$ is the complexity measure given by Fig. 2.4 and $\lambda_c$ is the Lagrangian multiplier that can be used to adjust complexity-video quality trade-off.

Taking the complexity into account, the cost function in (2.3) becomes

$$J(\texttt{mode}, \lambda_{\texttt{mode}}, \lambda_{mc}) = J(mode, \lambda_{\texttt{mode}}) + \lambda_{mc} C_{\texttt{mode}}(\texttt{mode}), \tag{2.7}$$

where $C_{mode}(mode)$ is the sum of the complexity measure of all motion vectors required in decoding the MB. This will also be used as the metric when evaluating result at the decoder.

Li et al. [39] proposed possible Qp-dependent values of $\lambda_c$ and $\lambda_{mc}$ as

$$\lambda_c = K_{\texttt{motion}} * ln(\lambda_{\texttt{motion}} + 1) \tag{2.8}$$

and

$$\lambda_{mc} = K_{\texttt{mode}} * ln(\lambda_{\texttt{mode}} + 1) \tag{2.9}$$

where $K_{\texttt{motion}}$ and $K_{\texttt{mode}}$ are used to obtain a good tradeoff between the coding efficiency and complexity.

After mode selection is done, the cost of motion refinement is compared to the cost of no motion refinement by evaluating and comparing the two cost functions:

$$J(\texttt{mode}', \lambda_{\texttt{mode}}, \lambda_{mc}) = J(\texttt{mode}', \lambda_{\texttt{mode}}) + \lambda_{mc} C_{\texttt{mode}}(\texttt{mode}'), \tag{2.10}$$

and

$$J(\texttt{!mr}, \lambda_{\texttt{mode}}, \lambda_{mc}) = J(\texttt{!mr}, \lambda_{\texttt{mode}}) + \lambda_{mc} C_{\texttt{mode}}(\texttt{!mr}), \tag{2.11}$$

where $\texttt{mode}'$ is the mode the MB would assume if motion refinement is carried out and !mr is the base layer skip mode where base layer motion is used. $C_{\texttt{mode}}(\texttt{!mr})$ is the cost of motion compensation with base layer motion information. In this way, decisions on whether to carry out motion refinement in the FGS enhancement layers are made in a rate-distortion-complexity optimized fashion.

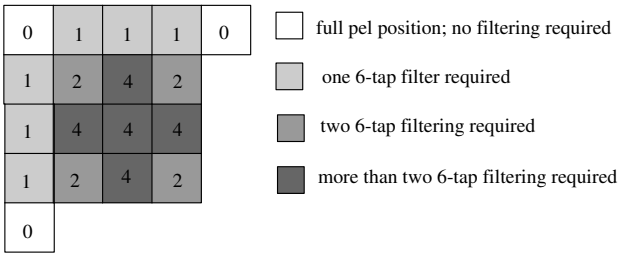



FIGURE 2.4: Sub-pixel positions and their assigned complexity

| Sequence | GOP Size |
|---|---|
| BUS,CITY,FOREMAN,HARBOUR,MOBILE | 16 |
| SOCCER,FOOTBALL | 8 |
| CREW | 4 |

TABLE 2.3: GOP sizes

## 2.8 Experimental Results

Our experiment was carried out on eight CIF sequences at 15fps: Foreman, Football, Soccer, City, Mobile, Bus, Harbour and Crew with encoder configurations defined by [32]. GOP structure with hierarchical B frames was used. GOP sizes defined by [32] are shown in Table 2.3. The value of Qp in base layer was set to 46 for all sequences unless otherwise stated. The search range was set to 96 and the maximum number of reference pictures in each reference list was 1. Encoded files were truncated to the desired bit rates with the accompanying bit stream extractor. Three FGS layers were encoded. Modifications were made on the JSVM software version 6.5. The experiments were carried out on a Pentium 4 3.2GHz machine with 1GB of RAM.

### 2.8.1 Encoder optimization with MB mode constraints

In this part of our experiment, we tested the impact on complexity and coding efficiency of the various MB mode constraints we described in the previous sections. The encoding time reduction brought about by the different encoder operation modes were collected; these times were compared against a FGS encoder with no motion refinement in the FGS layers and the results are recorded in Table 2.4. First, enhancement MB mode constraint (Section 2.6.1) and selective MB refinement to avoid refinement in successive co-located FGS MB (Section 2.6.2) were implemented (“Enhancement Layer Constraint" in Table 2.4).

| sequence | Full Mode Decision | Enhancement Layer Constraints | | base MB + enh MB constraint | all constraints | all constraints encoder fps |
|---|---|---|---|---|---|---|
| | | enh MB constraint | successive refinement constraint | | | |
| BUS | 6.32 | 2.47 | 5.22 | 1.88 | 1.56 | 0.17 |
| CITY | 5.55 | 1.94 | 4.56 | 1.51 | 1.43 | 0.23 |
| FOREMAN | 6.02 | 2.18 | 5.08 | 1.57 | 1.42 | 0.20 |
| SOCCER | 6.63 | 2.53 | 5.59 | 1.86 | 1.61 | 0.17 |
| CREW | 6.91 | 2.44 | 5.69 | 1.92 | 1.65 | 0.17 |
| HARBOUR | 8.17 | 2.32 | 6.61 | 1.99 | 1.68 | 0.24 |
| MOBILE | 5.72 | 1.98 | 4.67 | 1.44 | 1.24 | 0.28 |
| FOOTBALL | 7.31 | 2.84 | 6.13 | 2.00 | 1.67 | 0.13 |

TABLE 2.4: Number of times the complexity of an encoder with no motion refinement

Next, MB mode constraint in the base layer (with only 16x16 MB in the base layer) was implemented with enhancement MB mode constraint ("Base MB + Enh MB constraint" in Table 2.4).

Finally, all constraints were implemented, resulting in an encoder of lowest complexity. For this part of the experiment, $Qp_{motion}$ were kept the same as $Qp$ used during coefficient coding. The R-D curves are shown in Fig. 2.5(a) to Fig. 2.6(d) where curves obtained from encoder with all constraints and only enhancement layers constraints are shown.

To show that motion refinement is the most beneficial when the base layer is of low bit-rate, we carried out the experiment on three sequences with a base layer Qp of 40 (Figs. 2.7, 2.8 and 2.9). The gain of motion refinement is indeed less significant. The fast encoding algorithms described did not significantly penalize the coding efficiency.

While motion refinement with full mode search increased encoding time by more than 6 times over an FGS encoder with no motion refinement in the enhancement layers(Table 2.4), the scheme with the proposed encoder operation mode increased encoding time by around 1.5 times. Despite the dramatically reduced encoding time, the proposed scheme was able to produce nearly comparable coding performance.

### 2.8.2 Optimizing base layer motion information for higher bitrates

We tested different choices of Lagrangian multiplier in the base layer derived with $(Qp_{base} - 6), (Qp_{base} - 12)$ and $(Qp_{base} - 18)$. No motion refinement takes place in the enhancement layers. This technique allows us to bias the coding performance of the encoder towards a certain bitrate while compromising the performance at other bitrates. The effect of the choices of Lagrangian multiplier in the base layer on the coding performance at different bitrates is shown in Figs. 2.10, 2.12, 2.14 and 2.16.

| | Mode A | | | | Mode B | | | |
|---|---|---|---|---|---|---|---|---|
| Layer | mot refine | Qp | $Qp_{motion}$ | MB Modes allowed | mot refine | Qp | $Qp_{motion}$ | MB Modes allowed |
| Base | _ | $Qp_{base}$ | $Qp_{base} - 6$ | SKIP,16x16,16x8,8x16 | _ | $Qp_{base}$ | $Qp_{base} - 6$ | SKIP,16x16,16x8,8x16 |
| FGS 1 | no | $Qp_{base} - 6$ | _ | _ | no | $Qp_{base} - 6$ | _ | _ |
| FGS 2 | yes | $Qp_{base} - 12$ | $Qp_{base} - 12$ | Table 2.2 | yes | $Qp_{base} - 12$ | $Qp_{base} - 14$ | Table 2.2 |
| FGS 3 | yes | $Qp_{base} - 18$ | $Qp_{base} - 18$ | Table 2.2 | no | $Qp_{base} - 18$ | _ | _ |

TABLE 2.5: Two Encoder Operation Modes

| Sequence | Full Mode Decision | Mode A | Mode A fps | Mode B | Mode B fps |
|---|---|---|---|---|---|
| BUS | 6.32 | 1.68 | 0.16 | 1.03 | 0.26 |
| CITY | 5.55 | 1.19 | 0.28 | 0.71 | 0.47 |
| FOREMAN | 6.02 | 1.34 | 0.21 | 0.84 | 0.33 |
| SOCCER | 6.63 | 1.60 | 0.17 | 0.99 | 0.28 |
| CREW | 6.91 | 1.67 | 0.17 | 1.07 | 0.26 |
| HARBOUR | 8.17 | 1.63 | 0.24 | 0.94 | 0.43 |
| MOBILE | 5.72 | 1.18 | 0.30 | 0.75 | 0.47 |
| FOOTBALL | 7.31 | 1.86 | 0.11 | 1.13 | 0.19 |

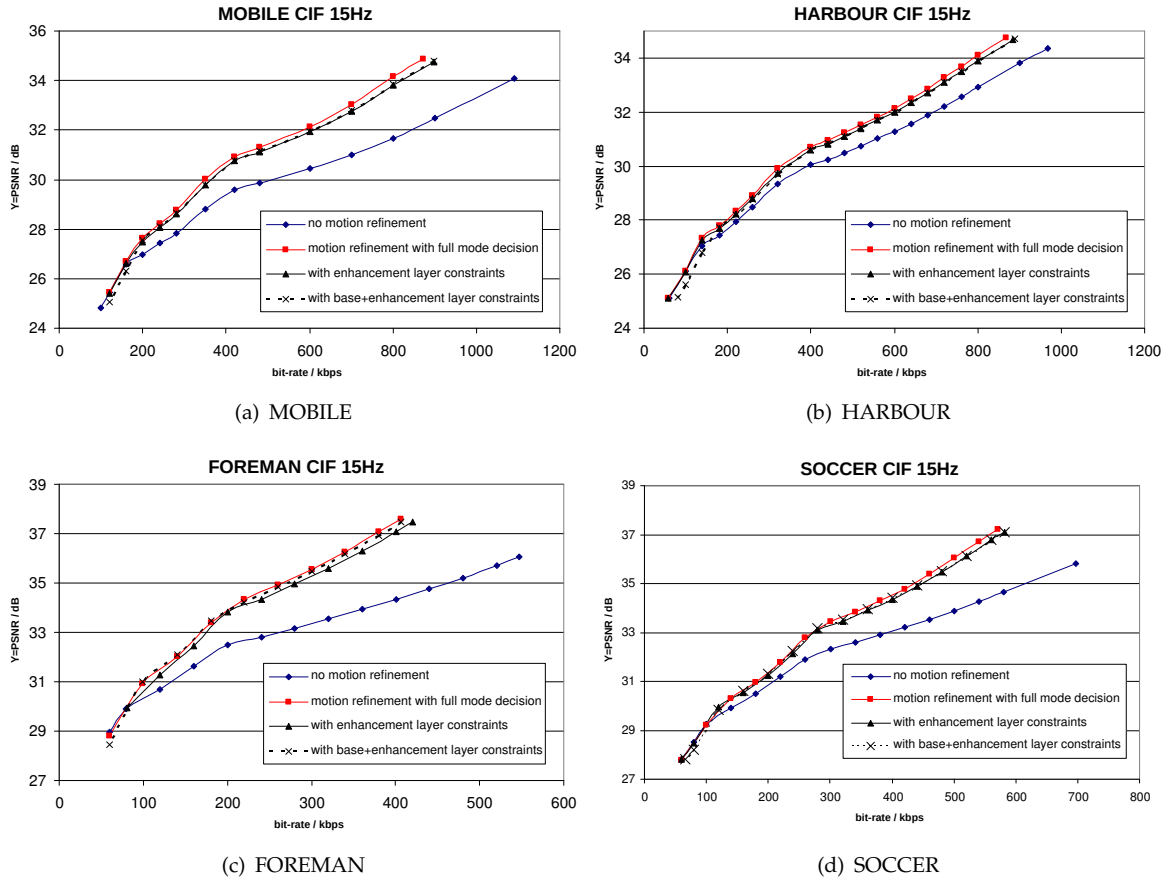
TABLE 2.6: Number of times the complexity of an encoder with no motion refinement (Modes A and B)

### 2.8.3 Two encoder operation modes

Among the three choices of motion estimation Qps in the base layer, the coding efficiency was improved by refining motion information in enhancement layers only in the case of $(Qp_{base} - 6)$. We define two encoder operation modes in Table 2.5 that encapsulate the proposed algorithms. In both modes, we derived the base layer Lagrangian multiplier with $(Qp_{base} - 6)$ and then selectively refine the motion information in the enhancement layers. The R-D curves for the sequences, Crew, Mobile, Bus and Foreman are shown in Fig. 2.11, 2.13, 2.15 and 2.17. The curve of full mode decision refinement was included in all graphs as a reference for comparison. The encoding times were compared against a FGS encoder with no motion refinement in the FGS layers and recorded in Table 2.6.

The results in Fig. 2.10, 2.12, 2.14 and 2.16 indicate that the performance that is achieved by decreasing the Lagrangian parameter (or the corresponding Qp value) and disabling motion refinement in the FGS layer is better than the results for FGS motion refinement at higher bitrates. It is possible that better coding performance can be achieved by encoder only techniques (by using a smaller Lagrangian multiplier) compared to motion refinement in enhancement SNR layers. However, a combination of techniques may be useful when it is undesirable to compromise the base layer coding efficiency. Also, using motion information that is optimized for a bitrate that is much higher than the base layer bitrate may restrict the lowest bitrate attainable by the base layer due to the larger amount of motion information that has to be carried.

By combining the techniques described in this chapter, the coding performance of the motion refined FGS scheme with exhaustive rate distortion computations can be achieved with far

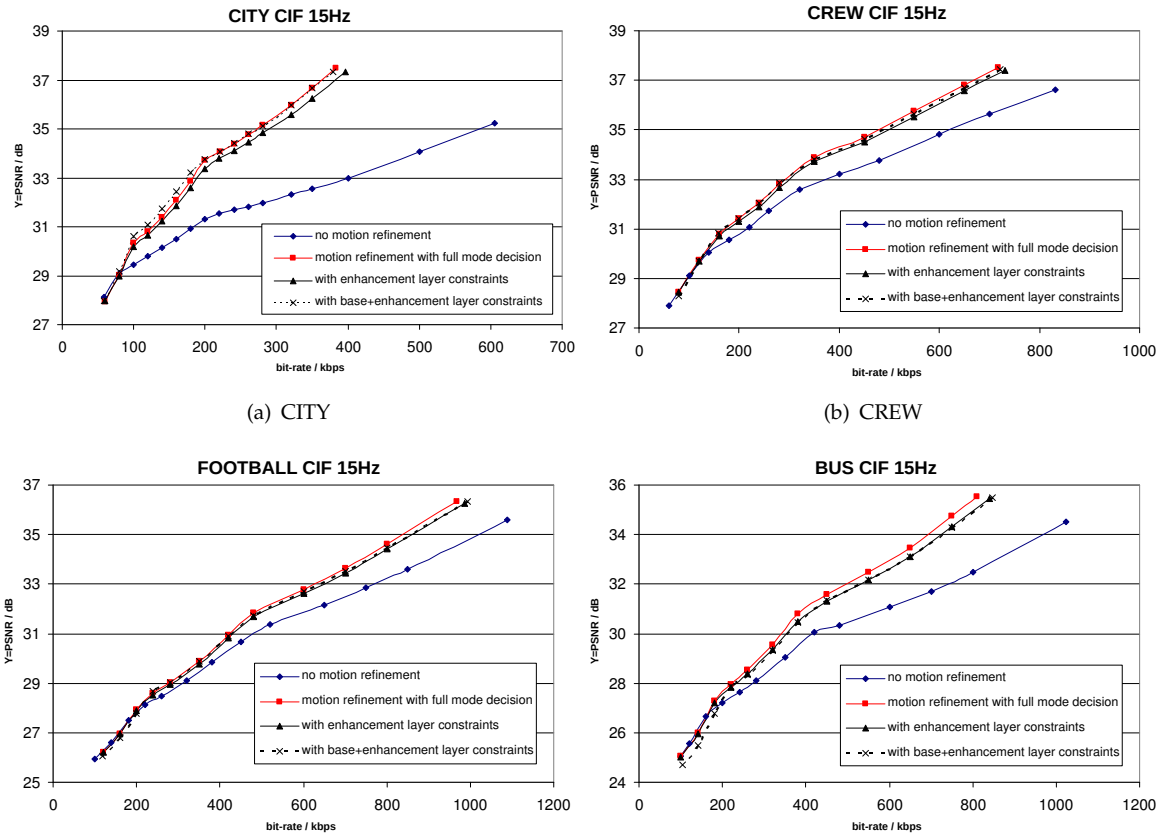



(a) MOBILE (b) HARBOUR

(c) FOREMAN (d) SOCCER

FIGURE 2.5: RD curves comparisons of encoders with different constraints

smaller number of rate distortion computations.




(a) CITY (b) CREW

(c) FOOTBALL (d) BUS

FIGURE 2.6: RD curves comparisons of encoders with different constraints

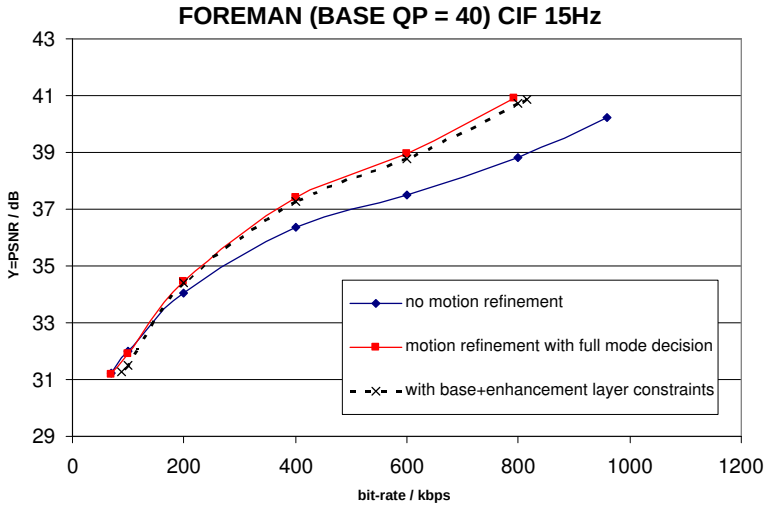



FIGURE 2.7: foreman (base Qp = 40)

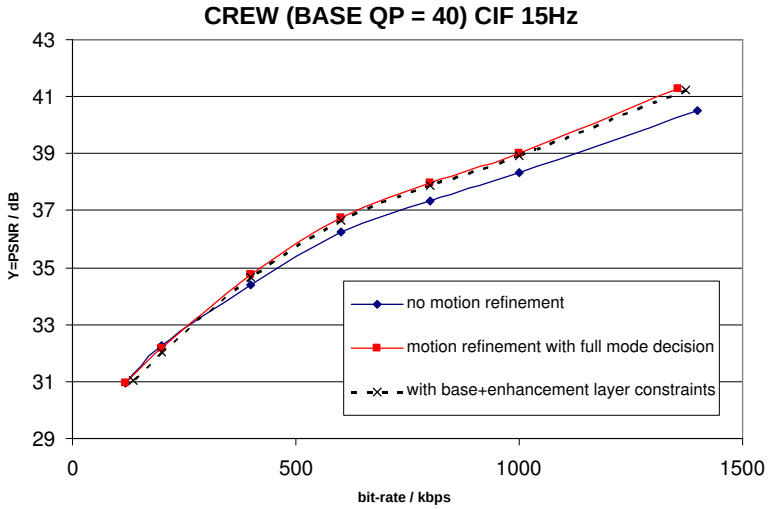



FIGURE 2.8: crew (base Qp = 40)

Motion refinement in enhancement layers can increase the coding efficiency of FGS video coding scheme. However, given the number of options available to an encoder, the encoding time for enhancement layer can increase dramatically. This chapter introduced a simplified scheme that substantially reduces the encoding time of enhancement layers while gains close to what is achievable by the exhaustive searching. This is achieved partly through enforcing a hierarchical partitioning of 16x16 areas in which an enhancement layer MB is more finely partitioned compared to the corresponding base layer MB during motion refinement.

By channeling part of the computational resource for motion estimation in the base layer into the enhancement layers for motion refinement, we were able to build an efficient FGS video encoding scheme at a much lower encoder complexity cost. Optimizing the base layer motion for lower Qp not only allows us to bias the coding performance of the encoder towards a certain bitrate while compromising the performance at other bitrates, it can also play a useful

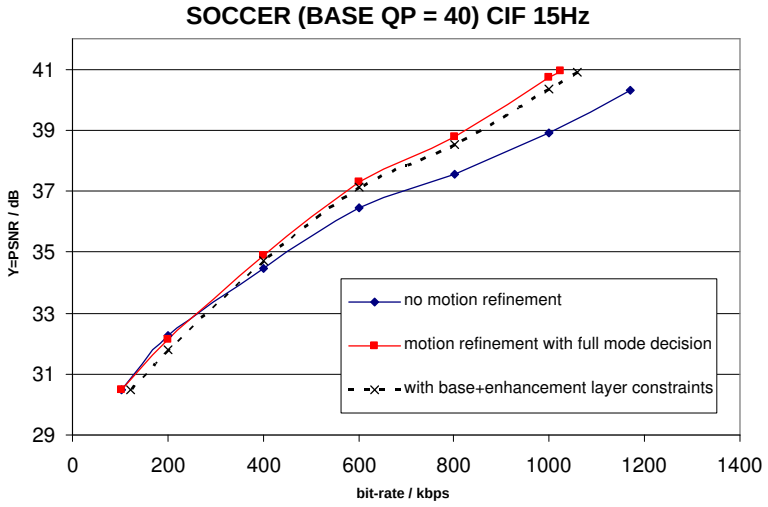



FIGURE 2.9: soccer (base Qp = 40)

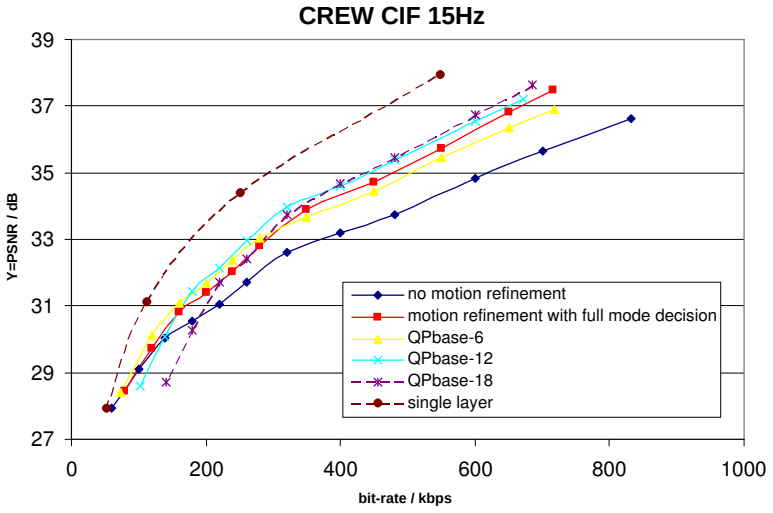



FIGURE 2.10: crew: base motion optimized for different Qp

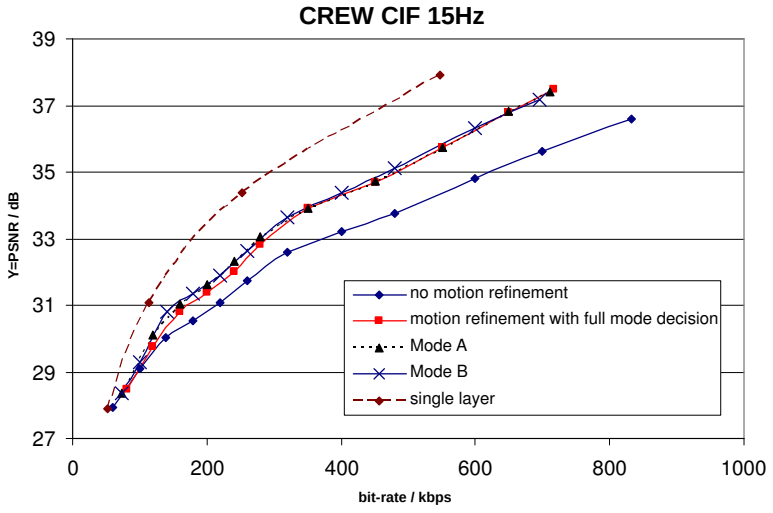



FIGURE 2.11: crew: Modes A and B

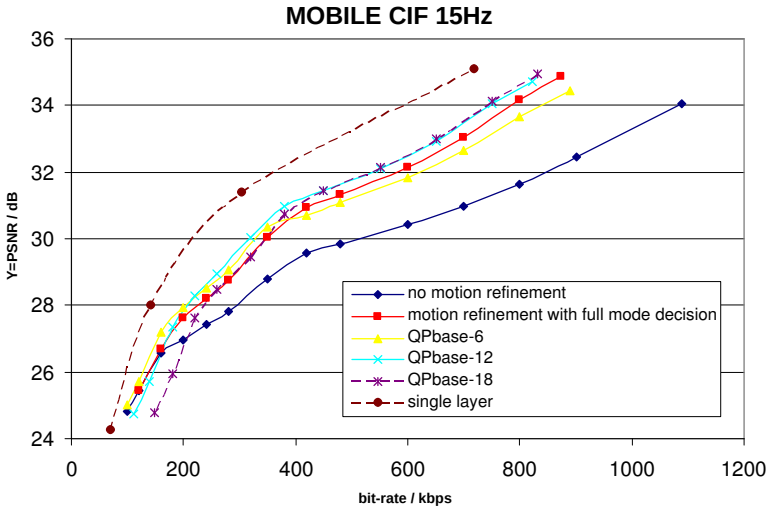



FIGURE 2.12: mobile: base motion optimized for different Qp

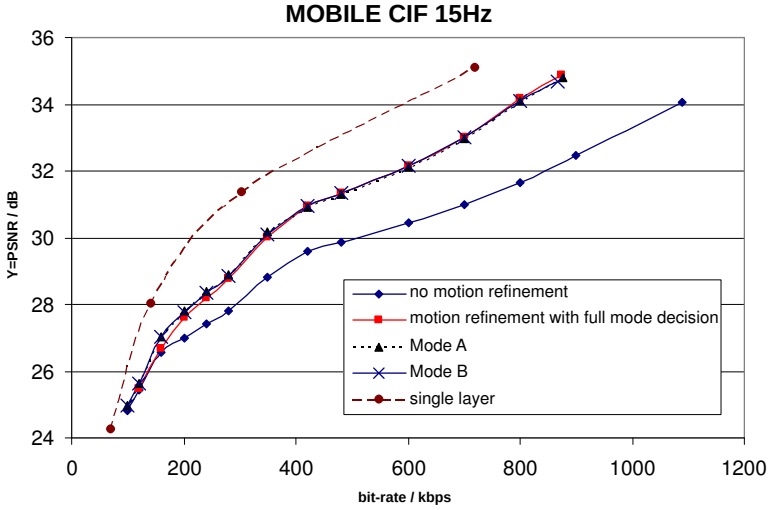



FIGURE 2.13: mobile: Modes A and B

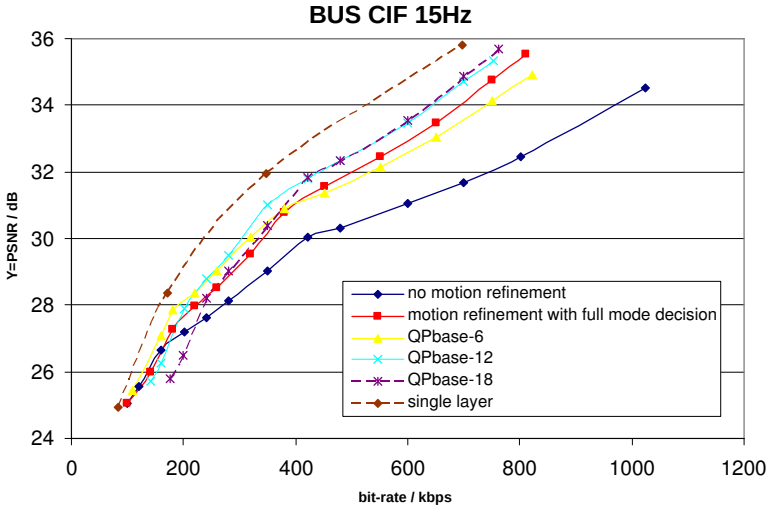



FIGURE 2.14: bus: base motion optimized for different Qp

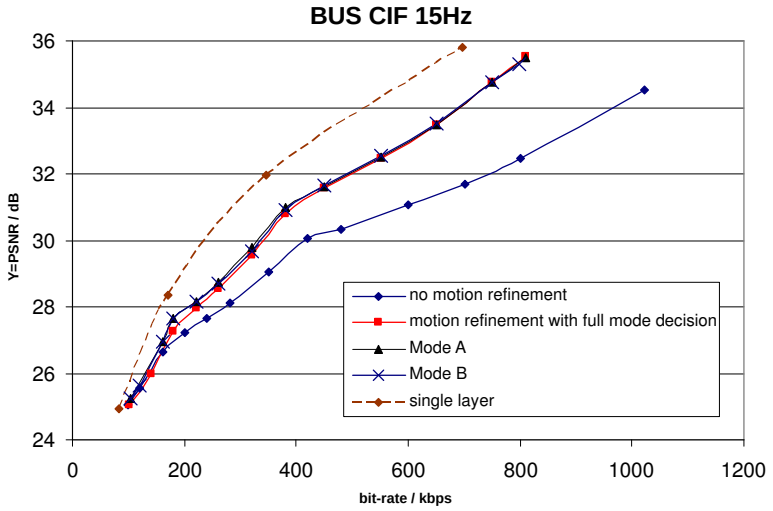



FIGURE 2.15: bus: Modes A and B

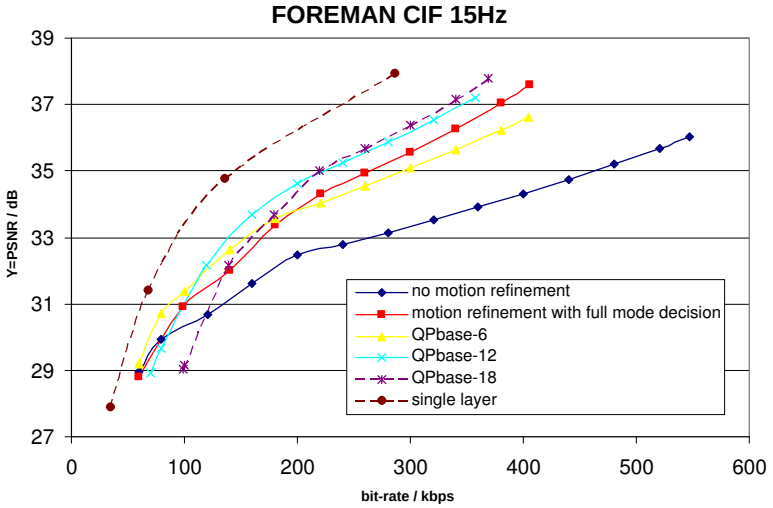



FIGURE 2.16: foreman: base motion optimized for different Qp

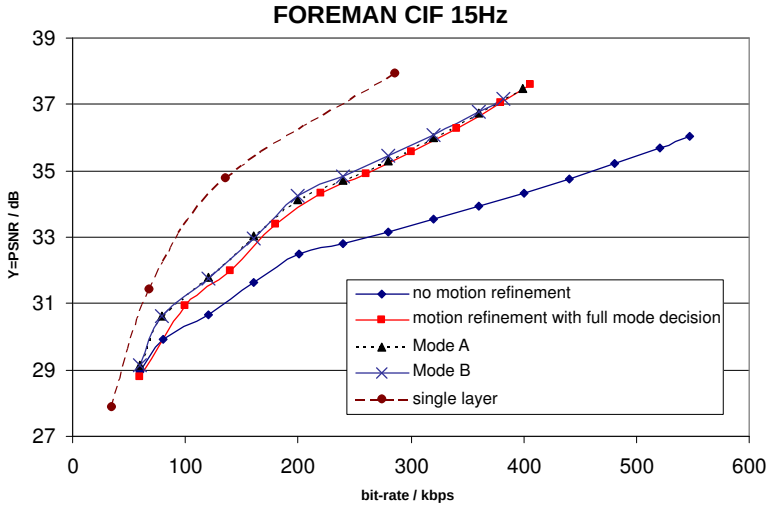



FIGURE 2.17: foreman: Modes A and B

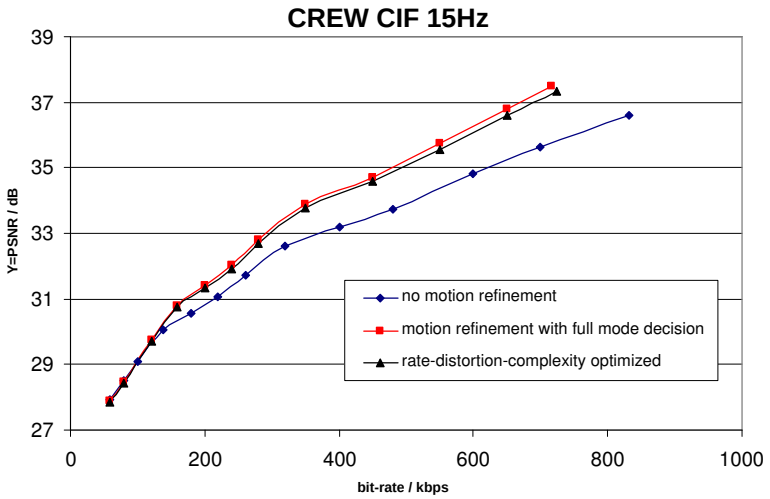



FIGURE 2.18: Rate-Distortion-Complexity optimized

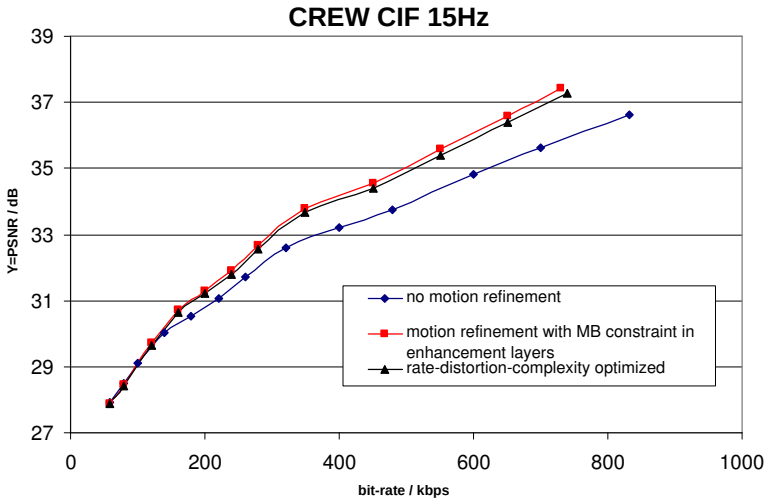



FIGURE 2.19: Rate-Distortion-Complexity optimized with enhancement layer MB mode constraint

role in the design of a fast FGS encoder. The various algorithms described allow the encoder to achieve a good tradeoff between motion and residue information throughout the extractable bitrates in a low complexity fashion.

By considering the interpolation complexity at the decoder when carrying out motion estimation in each layer and also when making motion refinement decision, the number of interpolations at the decoder can be reduced with an acceptable penalty to coding efficiency.

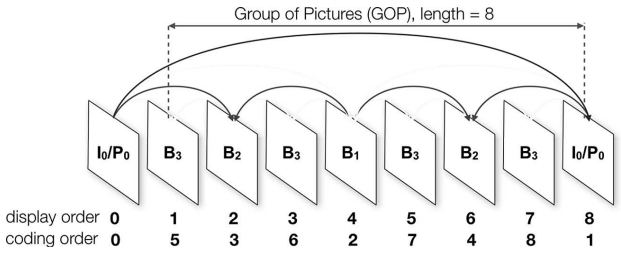



FIGURE 2.20: A hierarchal B-pictures coding structure with 4 temporal levels, indicated by the subscript of picture type (I, P or B).

## 2.9 LACING: An Improved Motion Estimation Framework for Scalable Video Coding

Temporal scalability in H.264/SVC video compression standard can be achieved with the hierarchical B-pictures (HB) structure. When performing motion estimation (ME) in the HB structure, the temporal distance between the current frame and reference frame(s) can be large (up to 32 frames apart). This limits the performance of fast search algorithm as larger search window is often necessary. Extensive experiments showed that popular fast suboptimal block ME algorithms are ineffective at tracking large motions across several frames. In the second part of this chapter, we propose a new framework called Lacing which integrates well with any fast block ME techniques to significantly improve the motion prediction accuracy in quality of the motion-compensated frame and also result in smoother motion vector fields with lower entropy.

Video compression can be achieved by reducing redundancies between video frames. Through blocked-based motion estimation (ME), a typical video encoder finds a set of motion vectors mapping the block that is being encoded to a block in the reference frame that best predicts its pixel values. The resulting motion vector fields are often correlated with object motions present in the video.

A best match for a $N \times N$ MB in the current frame can be found by searching exhaustively in the reference frame over a search window of $\pm R$ pixels. This amounts to $(2R+1)^2$ search points, each requiring $3N^2$ arithmetic operations to compute the SAD as the block distortion criterion. This is prohibitively high for software implementation. Many fast ME techniques have been proposed to reduce the number of search points using predefined search patterns and early termination criteria. Some well-known examples are: three-step [40], 2D logarithmic [41] and diamond [42] search. These fast techniques assume unimodal error surface; i.e., matching error increases monotonically away from the position of global minimum. When content motion is large or complex, the assumption of a unimodal error surface is no longer valid. Consequently, fast ME methods will produce false matches, thus leading to inferior quality motion-compensated frames that degrade coding performance.

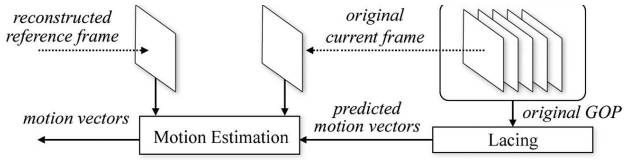



FIGURE 2.21: Lacing generates sets of predicted motion vectors, which are used to obtain more accurate motion estimation.

When the temporal distance between the current and reference frame is small, the inter-frame motion is also likely to be small. Hence, fast ME techniques work reasonably well in the conventional IPPP and IBBP coding patterns. However, the temporal distance can be much larger when performing scalable video coding that employs *hierarchical B-pictures* (HB) structure [43], which is supported by H.264/MPEG4-AVC [44] and adopted in the *joint scalable video model* (JSVM) [45]. In Fig. 2.20, frames at the lower temporal levels of the HB structure are motion estimated from reference frames that are temporally further apart. Larger inter-frame motion can be expected at lower temporal levels. The problem is further aggravated when the Group of Pictures (GOP) size is large. Fast ME algorithms, which are very effective for motion estimation over relatively small motion search ranges, can become ineffective when applied in the HB structure. Nevertheless, it is still desirable to use fast ME methods for their speed and simplicity.

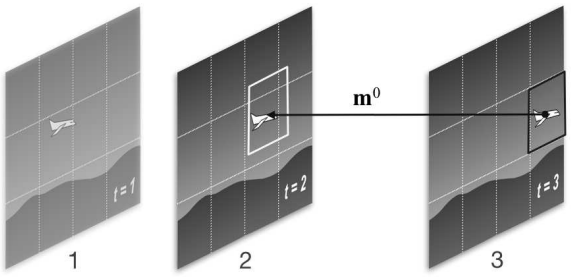
Here, we propose a framework called *Lacing* that integrates seamlessly with existing fast ME methods and improves their motion prediction accuracy when employed in the HB structure (or any coding structures where there exists intermediate frames between current and reference frames) by extending their effective motion search range through successive motion vector interpolation along the MB's motion trajectories across the frames within the GOP. The Lacing framework is also motivated by observations that rigid body motions produce continuous motion trajectories spanning a number of frames across time. By exploiting these motion characteristics, Lacing helps to progressively guide the motion prediction process while locating the 'true' motion vector even across a relatively large temporal distance between the current and reference frames.

## 2.10 Lacing Framework

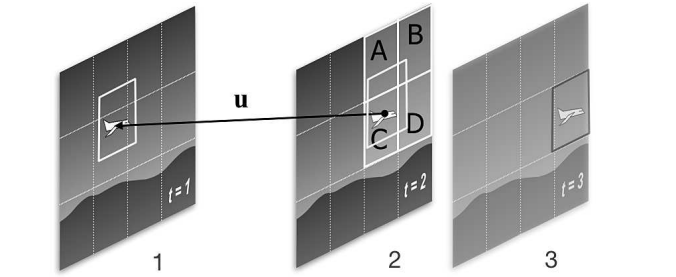
Having observed the motion continuity of rigid body motions across frames, the proposed Lacing framework exploits these strong temporal correlations in the motion vector fields of neighbouring frames, such that:

$$\mathbf{M}_{t,t-2}(\mathbf{p}) \approx \mathbf{M}_{t,t-1}(\mathbf{p}) + \mathbf{M}_{\substack{t-1,\\ t-2}}(\mathbf{p} + \mathbf{M}_{t,t-1}(\mathbf{p})) \tag{2.12}$$

where $\mathbf{M}_{t_1,t_0}$ denotes the set of motion vectors of current frame $f(t_1)$ with reference frame $f(t_0)$ and, $\mathbf{M}_{t_1,t_0}(\mathbf{p})$ represents the motion vector of MB positioned at $\mathbf{p}$ in the current frame




(a) Step 1: Equation (2.17) determines $\mathbf{m}^0$ and Equation (2.16) computes $\mathbf{p}_1$.




(b) Step 2: $\mathbf{m}^1 = \mathbf{p}_2 - \mathbf{p} = \mathbf{m}^0 + \mathbf{u}$, as in Equation (2.15).

FIGURE 2.22: Example of motion estimating current frame 3 from reference frame 1 by applying Lacing using Equation (2.15)–(2.17).

$f(t_1)$. Generally for $(t_1 - t_0) > 1$, $\mathbf{M}_{t_1,t_0}(\mathbf{p})$ can be approximated by $\mathbf{m}^{t_1-t_0-1}$ using the following iterative equation,

$$\mathbf{m}^j = \mathbf{m}^{j-1} + \mathbf{M}_{\substack{t_1-j, \\ t_1-j-1}}(\mathbf{p} + \mathbf{m}^{j-1}), \tag{2.13}$$

with initial condition

$$\mathbf{m}^0 = \mathbf{M}_{t_1,t_1-1}(\mathbf{p}). \tag{2.14}$$

It is noted that the updating term in equation (2.13) is a motion vector from $f(t_1 - j)$ to $f(t_1 - j - 1)$, which is only across a unit temporal interval. Thus, the updating motion vector can be computed using fast (or small search range) ME methods. This contrasts with the direct computation of $\mathbf{M}_{t_1,t_0}(\mathbf{p})$, which would otherwise require the estimation of motion vector over a large search range if $t_1 - t_0$ is large.

In each iteration of equation (2.13), the MB at $\mathbf{p} + \mathbf{m}^{j-1}$ has to be motion estimated. Using the exhaustive method with $\pm v$ motion search range, each MB require an average of $(t_1 - t_0)(2v + 1)^2$ search points. For a GOP of $T$ frames and with $1 + \log_2 T$ temporal levels in the HB structure, each MB will require an average of $(1 + \log_2 T)(2v + 1)^2$ search points. The following algorithm outlines the steps to reduce the average number of search points to $(2v + 1)^2$ per MB.

### 2.10.1 The Algorithm

For $t_0 \neq t_1$, $\mathbf{M}_{t_1,t_0}(\mathbf{p})$ is approximated by $\mathbf{m}^{|t_1-t_0|-1}$ from the following iterative equations:

$$\mathbf{m}^j = \mathbf{m}^{j-1} + \mathbf{u}\left(\mathbf{M}_{\substack{t_1-s\cdot j,\\ t_1-s\cdot(j+1)}}, \mathbf{p}_j\right) \tag{2.15}$$

$$\mathbf{p}_j = \mathbf{p} + \mathbf{m}^{j-1} \tag{2.16}$$

with $s = \mathrm{sgn}(t_1 - t_0)$ and the initial condition

$$\mathbf{m}^0 = \mathbf{M}_{t_1,t_1-s\cdot 1}(\mathbf{p}). \tag{2.17}$$

The updating vector function $\mathbf{u}$ in equation (2.15) is a motion vector at $\mathbf{p}_j$ interpolated ( bilinear is used in this work) from the neighboring motion vectors:

$$\begin{aligned}
&\mathbf{M}_{\substack{t_1-s\cdot j,\\ t_1-s(j+1)}}(N\lfloor \mathbf{p}_j/N \rfloor),\\
&\mathbf{M}_{\substack{t_1-s\cdot j,\\ t_1-s(j+1)}}(N(\lfloor \mathbf{p}_j/N \rfloor + [1,0])),\\
&\mathbf{M}_{\substack{t_1-s\cdot j,\\ t_1-s(j+1)}}(N(\lfloor \mathbf{p}_j/N \rfloor + [0,1])),\\
&\mathbf{M}_{\substack{t_1-s\cdot j,\\ t_1-s(j+1)}}(N(\lfloor \mathbf{p}_j/N \rfloor + [1,1])).
\end{aligned}$$

Equations (2.15)–(2.17) forms the core computing steps in our proposed *Lacing* framework, which is outlined in Algorithm 1 for motion estimating frames in the HB structure.

Unlike equation (2.13), no motion estimation is required when evaluating the updating vector in equation (2.15) since $\mathbf{M}_{t,t\pm1}$ can be pre-calculated (see step 1–2 in Algorithm 1). We only need to access $\mathbf{M}_{t,t\pm1}$ at fixed MB positions.

### 2.10.2 Complexity Analysis

When motion estimation is used with Lacing, the computation overheads are attributed to the following processes:

- ME is performed during the pre-calculation (step 1–2) and the predicted motion vectors refinement (step 12) stages of Algorithm 1. Depending on the actual ME strategy used, Lacing can introduce up to an additional 2 times the number of search points per MB. This is acceptable since fast ME techniques already have very low average search points to begin with.
- Interpolating the motion vectors in Equation (2.15) requires only a relatively small computation. In the bilinear interpolation case, $2 \times (12\texttt{MULS}+6\texttt{ADDS})$ is required for each

**Algorithm 1**: Lacing framework for HB structure

**Input**: $f(0)$, first frame in sequence or last frame from previous GOP.
**Input**: $\{f(1), f(2), \ldots, f(T)\}$, GOP of length $T$.
**Output**: $\hat{\mathbf{M}}$ , sets of predicted motion vectors

```
Compute {M_{t,t-1} : 1 ≤ t ≤ T};
Compute {M_{t,t+1} : 1 ≤ t < T};
for t ← 1 to T do
    D ← temporal distance of f(t) from its reference;
    if D>1 then
        foreach MB at p in f(t) do
            M̂_{t,t-D}(p) ← approx. M_{t,t-D}(p) using Equations (2.15)–(2.17);
            if temporal level of f(t) >0 then
                M̂_{t,t+D}(p) ← approx. M_{t,t+D}(p) using Equations (2.15)–(2.17);
            end
        end
        Refine M̂_{t,t-D}(p) and M̂_{t,t+D}(p) with ME;
    else
        M̂_{t,t-D}(p) ← M_{t,t-1}(p);
        M̂_{t,t+D}(p) ← M_{t,t+1}(p);
    end
end
```

MB. This is insignificant, compared to $N^2$ABS$+(2N^2-1)$ADDS required to compute the SAD of $N \times N$ MB at each search point.

Using the exhaustive method with a search range of $\pm v$ pixels, and applying Lacing to a HB-structured GOP of $T$ frames and $1 + \log_2 T$ temporal levels requires an average of $(4 - 3/T)(2v+1)^2$ search points, or $2(2v+1)^2$ search points without the refinement step 12 in Algorithm 1.

## 2.11 Experimental Results

In order to investigate the effectiveness of the proposed Lacing framework, we compared two popular fast block ME algorithms (the Diamond search [42] and TZ-search used in the JSVM software [46]) against their corresponding enhanced counterparts when integrated within the proposed Lacing framework. The exhaustive full search method is used as a benchmarking reference.

With reference to Table 2.7, the following test criteria are used to compare the ME performance:

- *Mean search points per MB* (MSP): This is proportional to the computation complexity, and hence computing time, required by the ME strategy.

TABLE 2.7: Performance comparison of various motion estimation techniques on different video sequences

| | Stefan | | | | | | | | City | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Luminance Y PSNR (dB) | | | | | | | | Luminance Y PSNR (dB) | | | | | |
| | | | Temporal level | | | | | | | | Temporal level | | | | | |
| | MSP | ACL | 0 | 1 | 2 | 3 | 4 | Avg. | MSP | ACL | 0 | 1 | 2 | 3 | 4 | Avg. |
| $ES_{32}$ | 7010 | 5.50 | 19.41 | 22.66 | 25.17 | 26.26 | 27.13 | 25.23 | 7010 | 5.71 | 23.26 | 28.20 | 30.43 | 31.62 | 32.31 | 30.14 |
| $ES_{16}$ | 1908 | 5.40 | 18.67 | 20.93 | 22.48 | 26.05 | 27.11 | 24.37 | 1908 | 5.80 | 22.14 | 24.46 | 29.11 | 31.59 | 32.30 | 29.11 |
| $TZ_{32}$ | 568 | 5.63 | 19.17 | 22.27 | 24.69 | 25.94 | 27.07 | 24.99 | 647 | 6.29 | 23.06 | 27.36 | 29.62 | 31.02 | 32.01 | 29.70 |
| $TZ_{32}$-L | **617** | **5.47** | **20.20** | **23.86** | **25.09** | **26.23** | **27.07** | **25.50** | **634** | **5.86** | **24.14** | **28.29** | **30.22** | **31.48** | **32.01** | **30.24** |
| $TZ_{16}$ | 233 | 5.51 | 18.46 | 20.56 | 22.15 | 25.78 | 27.04 | 24.15 | 264 | 6.22 | 21.97 | 24.12 | 28.39 | 31.03 | 31.99 | 28.75 |
| $TZ_{16}$-L | **272** | **5.35** | **19.76** | **23.43** | **24.65** | **26.15** | **27.04** | **25.28** | **287** | **5.73** | **23.54** | **27.42** | **29.87** | **31.45** | **31.99** | **29.94** |
| DS | 37 | 5.37 | 16.56 | 18.96 | 20.04 | 21.95 | 25.79 | 22.00 | 46 | 6.50 | 20.81 | 22.41 | 24.45 | 27.81 | 31.18 | 26.74 |
| DS-L | **41** | **5.27** | **19.99** | **23.53** | **24.76** | **25.84** | **25.79** | **24.83** | **45** | **5.74** | **24.19** | **28.16** | **29.89** | **31.14** | **31.18** | **29.85** |

| | Foreman | | | | | | | | Mobile Calendar | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Luminance Y PSNR (dB) | | | | | | | | Luminance Y PSNR (dB) | | | | | |
| | | | Temporal level | | | | | | | | Temporal level | | | | | |
| | MSP | ACL | 0 | 1 | 2 | 3 | 4 | Avg. | MSP | ACL | 0 | 1 | 2 | 3 | 4 | Avg. |
| $ES_{32}$ | 7010 | 5.31 | 27.47 | 29.29 | 30.96 | 33.01 | 34.74 | 32.38 | 7010 | 5.33 | 18.36 | 22.53 | 23.46 | 24.18 | 24.55 | 23.44 |
| $ES_{16}$ | 1908 | 5.14 | 26.05 | 28.51 | 30.72 | 32.92 | 34.72 | 31.90 | 1908 | 5.38 | 15.80 | 21.36 | 23.41 | 24.16 | 24.54 | 22.69 |
| $TZ_{32}$ | 366 | 5.47 | 26.99 | 29.06 | 30.85 | 32.88 | 34.60 | 32.17 | 406 | 5.58 | 17.76 | 21.64 | 22.42 | 23.86 | 24.50 | 23.02 |
| $TZ_{32}$-L | **337** | **5.29** | **27.24** | **29.28** | **30.99** | **32.93** | **34.60** | **32.28** | **291** | **5.48** | **18.30** | **22.31** | **23.29** | **24.13** | **24.50** | **23.36** |
| $TZ_{16}$ | 174 | 5.31 | 25.56 | 28.34 | 30.58 | 32.76 | 34.57 | 31.67 | 192 | 5.51 | 15.53 | 20.58 | 22.61 | 23.86 | 24.50 | 22.38 |
| $TZ_{16}$-L | **186** | **5.16** | **26.29** | **28.81** | **30.82** | **32.89** | **34.57** | **31.97** | **83** | **5.35** | **16.86** | **22.02** | **23.28** | **24.11** | **24.50** | **23.00** |
| DS | 38 | 5.36 | 23.48 | 27.53 | 30.30 | 32.63 | 34.43 | 30.84 | 38 | 5.59 | 14.08 | 16.75 | 19.38 | 23.13 | 24.48 | 20.87 |
| DS-L | **40** | **5.12** | **26.58** | **28.95** | **30.83** | **32.91** | **34.43** | **32.02** | **38** | **5.06** | **17.81** | **22.24** | **23.20** | **24.02** | **24.48** | **23.21** |

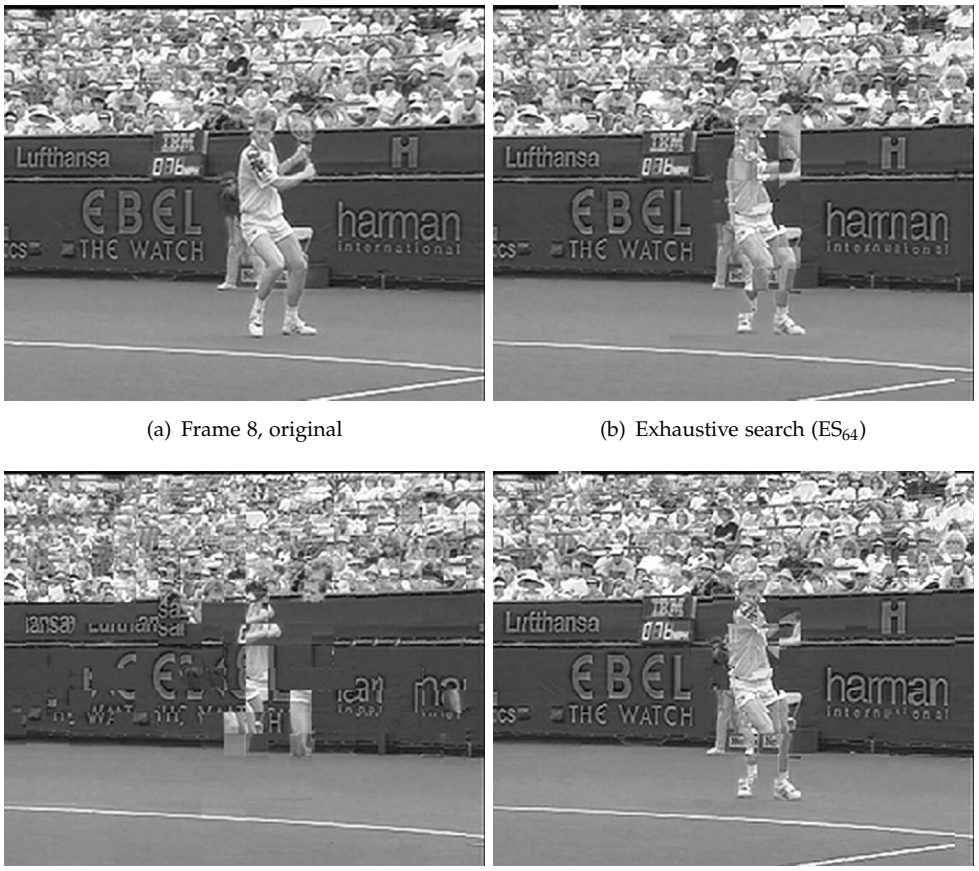
(ES: exhaustive search , TZ: TZ search [46], TZ-L: TZ with Lacing , DS: Diamond search [42], DS-L: DS with Lacing; Search range is denoted by number in subscript.) It is evident that integrating the proposed Lacing framework with fast sub-optimal block matching algorithms can significantly improve ME accuracies, with the quality of the motion compensated sequence improved by as high as 3.11 dB at only a fraction of the computation cost of exhaustive search.

- *Peak signal-to-noise ratio* (PSNR): This measures the quality of the motion-compensated frames, which is dependent on the accuracy of the ME method. A low value means poor frame prediction and significant errors. For color sequences, we only show the PSNR luminance data due to space constraint.

- *Average code-length of motion vectors* (ACL): This provides an estimate on the average code length (bits per motion vector) required to code the motion vectors. As in H.264/SVC, motion vectors in each frame are median predicted and the differential motion vectors are coded using exp-Golomb codes.

The following standard test sequences at CIF resolution are used: *Stefan*, *City*, *Foreman* and *Mobile Calendar*. GOP size is fixed at 16 frames with a HB structure of 5 temporal levels. In the experiments, each $16 \times 16$ luminance MBs is motion estimated with integer-pel precision. Scaled motion vectors are used for corresponding chroma MBs when reconstructing the motion-compensated pictures.

Table 2.7 summarizes the performance of various ME techniques over the aforementioned test criteria. The Lacing framework gives significant quality gain in the motion compensated pictures: an average of 2.36 dB gain over all sequences for the Diamond-Lacing search strategy;




(a) Frame 8, original (b) Exhaustive search ($ES_{64}$)

(c) Diamond search (d) Diamond-Lacing

FIGURE 2.23: Examples of bi-directionally motion-compensated Frame 8 using various motion estimation methods with reference frames 0 and 16.

0.38 dB and 0.81 dB gain for TZ-Lacing with search range $\pm 32$ and $\pm 16$ respectively. The largest quality gain are observed in frames from the lower temporal levels, where the gain of individual frame can be up to 5.75dB.

Lacing also improves the compressibility of motion vectors by 3.7% to 6.9%, averaged over all sequences for each Lacing variant ME method. Using Lacing with fast search algorithms significantly improves prediction performance at the cost of a modest increment (3.5% to 7.0%) of search points. In some cases, Lacing reduces the average number of search points required. For example in the *Mobile Calendar* sequence, $TZ_{32}$-Lacing only requires 291 average search points, compared to 406 search points by $TZ_{32}$. This is because Lacing only compute ME between directly adjacent frames; thus, the local minimum can be found quickly.

Fig. 2.23 shows examples of motion compensated frames of the *Stefen* sequence, which consists of a panning background and a moving subject in the foreground. The objective here is to perform bi-directional motion estimation for Frame 8 in Fig. 2.23(a) using reference frames 0 and 16. In figure 2.23(c), it is evident that the fast ME method, Diamond search, is unable to give a reasonable prediction of frame 8 due to large inter-frame displacement. By integrating

Diamond search into the Lacing framework, the improvement in ME accuracy is obvious in Fig. 2.23(d). The lower motion vector field entropy and improved motion compensated frame will lead to better video compression performance too.

## 2.12 Remarks

Two encoders working within the specification of a particular video codec can display vastly different performances. In terms of encoding speed, an encoder with better decision-making algorithms can attain comparable coding performance compared with one operating with exhaustive mode decisions and motion searches at significantly reduced computation cost.

In this chapter, fast encoding algorithms were proposed for encoding in two scenarios: encoding videos with multiple quality layers and encoding a hierarchically arranged GOP.

During the encoding of layered video, cross-layer considerations during encoding can improve the coding performance of encoder and reduce the number of operations required to attain good performance. This possible encoder speed up is useful as exhaustive mode decisions in all layers can dramatically increase the complexity of the encoder. The proposed encoding scheme achieves good performance in all layers while incurring the computation cost of a single layer encoder.

The application of hierarchical B-pictures structure in the H.264/SVC video coding standard has introduced the challenge for effective motion estimation (ME) across frames with much larger temporal distance of up to 32 frames apart. Popular fast sub-optimal block ME algorithms, such as Diamond search, are very efficient for relatively small motion search ranges but perform poorly when estimating such larger motion. The proposed Lacing framework can integrate seamlessly with existing fast ME methods to extend their effective search range by tracing motion trajectories. Experiments showed that Lacing yield significantly better motion prediction accuracy by as high as 3.11 dB gain in quality and give smoother motion vector fields that can be coded more efficiently.

Of course, the encoder power consumption or complexity can only be reduced to a certain extent without hurting coding performance. Ideally, fast decision algorithms can reduced the complexity of the encoder to the desired level. However, when computational resource is limited, the complexity of the encoder may need to be further reduced. In Chapter 3, the method of achieving complexity-coding performance trade-off will described.

# Chapter 3

# Complexity Scalable Encoding

In Chapter 2, we suggested several methods of reducing the complexity of a video encoder. Ideally, the complexity of the encoder should be reduced to within the system constraints without compromising the coding performance. When the complexity of the encoder has to be reduced beyond what is possible with fast decision algorithms, a technique that allows the control of performance-complexity trade-off would be useful to ensure that the encoder makes the best use of available resource. In this chapter, we will describe an encoding scheme that allows complexity control with a single parameter.

The state-of-the-art H.264/AVC video coding standard enables significantly better video compression performance as compared to previous coding standards [36][44]. Though using the numerous new coding features in H.264 optimally will lead to good coding performance, it also increases the computational complexity of the encoder. Hence, designing a real-time H.264/AVC encoder with good coding performance poses a significant challenge.

A typical implementation of an encoder can be computationally complex for a few reasons: a large number of SAD operations carried out during motion searches, interpolations for subpixel motion estimation and transform and inverse transform operations during the reconstruction of encoded pictures for subsequent prediction. Any algorithms that can reduce the number of these operations or implementation techniques that can speed them up can conceivably increase the speed of the encoder.

During block-based video encoding, the encoder has to evaluate a number of possible coding modes[1] to determine the optimal representation for each MB. In order to fully exploit the features to achieve optimal rate-distortion (R-D) performance, the encoder has to check all possible coding modes and select the best set of coding parameters. The complexity of this exhaustive approach has motivated a host of fast algorithms [34][47][48] that provide encoding complexity reduction with sub-optimal R-D performance. These fast algorithms work by evaluating only a subset of all possible modes; coding modes that are judged to be less

[1] methods to partition, predict and encode the data.

probable are omitted from R-D operations. Fast search algorithms during block-based motion estimation also play a big part in reducing the complexity of the encoding process. These algorithms reduce the number of search points by following a pre-defined search path that can be shown to result in good prediction [42], using stop criteria during searches [49] or using good starting points for searches [2]. Algorithms that speed up sub-pel motion estimation include [50] and [51].

Despite the availability of various fast algorithms for H.264 encoder, it is difficult to design an optimized H.264 encoder given a target computing platform. It is not obvious how the different fast encoding algorithms can be used to control the encoding process to achieve a good arbitrary trade-off between complexity and coding efficiency. Complexity scalability, where the computational complexity of an encoder can be scaled with a trade-off in coding performance, is a useful tool. When computational resource is limited but a fast implementation of the encoder is required, the complexity of the encoder can be scaled down to ensure that encoding can be done on time. Real-time encoding is required for applications such as live broadcast, surveillance or video communication. Since these applications may be built on a wide variety of computing platforms, to make full use of computational resource while ensuring that encoding completes on time will be difficult without an effective complexity scalable solution.

Some encoder complexity scalable schemes have previously been proposed. In [52], dynamically parameterized architectures are proposed for motion estimation and discrete cosine transform. These enable the video encoding process to gracefully degrade in power-constraint environments. In [53], the complexity of H.263+ encoding is controlled by pre-determining the proportion of `SKIP` [2] and restricting the search range during motion estimation and then assigning more SAD computations to regions that are predicted to have high motion content. Decisions to skip or code MBs are made based on the prediction residue after motion compensated prediction. Ismaeil et al [54] achieves complexity control by empirically determining a set of encoder operation modes that gives different complexity-performance trade-off. In these approaches, the complexity of the encoder is controlled at a low level (e.g., the building/processing blocks of the encoder such as the motion estimation and the DCT transform) using several parameters. This results in a multi-dimensional complexity-distortion (at fixed rate) surface that can be inconvenient for encoder designers to find an optimal set of parameters for the target platform/application. Works that describe computationally scalable motion estimation algorithms also include [55] and [56]. With these methods, the extent to which the complexity of the encoder can be scaled down is restricted; if the motion estimation module comprises 40% of the total encoding cost, the complexity of the encoder cannot be scaled back by more than 40%. The work in [57] controls the complexity of H.264 encoding by controlling the proportion of SKIP MBs through prediction mechanisms that try to identify SKIP MBs prior to coding. Attempts to control encoder complexity in [58] and [59] require prior knowledge about the relative complexity and coding performance of available encoding tools.

[2] `SKIP` MBs are computationally less expensive to code.

This work introduces a singularly-parameterized complexity scalable rate-distortion framework for H.264/AVC encoders. This high-level approach can also serve as a complexity scalable platform for other existing complexity reduction algorithms and techniques.

The rest of the chapter is organized as follows: Section 3.1 discusses the decision-making process of a typical encoder through R-D optimization; Section 3.2 introduces our proposed complexity scalable R-D optimization scheme and Section 3.3 describes the application of the algorithm to scalable video coding. We also discuss the possibility of building a power-aware encoding scheme in Section 3.4. Section 3.5 presents the experimental results before Section **??** concludes the paper.

## 3.1 Rate-Distortion Optimization in H.264/AVC

A typical video encoding scheme consists of various coding methods (or *modes*) to accommodate different properties of video data. Within each video frame, the nature of the video data is not uniform, i.e., there are texture-filled, edge-filled and homogeneous regions. The levels of motion activity (compared to reference frames) can also vary. Therefore, it is most effective for the video encoding process to partition a video frame into macroblocks and find the best coding modes for each of them.

The large number of coding options available for the encoding of each MB means that the encoder has to intelligently choose the combination of coding modes that leads to the best compression. This is certainly a time-consuming and challenging optimization task.

The use of variable block-sizes can significantly improve coding performance. Using smaller block size requires the coding of more header information but can provide better motion compensated prediction, especially when coding region with high activity. On the other hand, the `SKIP` mode and large block sizes are effective for coding stationary regions with little activity across adjacent frames. To fully exploit the benefits of variable block-size motion compensation, the encoder needs to adaptively choose the most effective partition size during motion estimation. H.264/AVC provides several MB modes for motion compensated prediction. Each mode corresponds to a specific partition of a 16x16 MB. The standard supports modes that partition a MB into 16x16, 16x8, 8x16 and 8x8 luminance samples. Each 8x8 sub-block may be further partitioned into blocks of 8x8, 8x4, 4x8, 4x4 luminance samples. The use of rate-distortion optimization for decision making during video encoding has been well studied in [60][61].

Suppose that $Qp$ is the quantization parameter. During motion estimation, the encoder selects a motion vector, $\widetilde{\mathbf{m}} = [m_x, m_y]$ that minimizes the following cost function:

$$J(\widetilde{\mathbf{m}}, \lambda_{\mathtt{mot}}) = SAD + \lambda_{\mathtt{mot}} R(\widetilde{\mathbf{m}}) \tag{3.1}$$

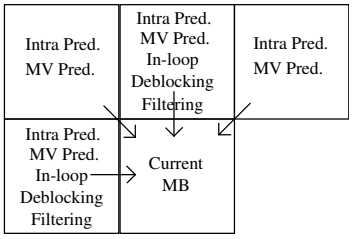



FIGURE 3.1: MB dependency in H.264: Motion vectors (MVs) of neighbouring MBs are required for prediction of MVs of current MB. Reconstructed values are needed for intra prediction and deblocking filter.

where SAD is the *sum of absolute differences* between the original signal and predicted signal, $R(\widetilde{\mathbf{m}})$ is the number of bits required to code $\widetilde{\mathbf{m}}$, and

$$\lambda_{\texttt{mot}} = 0.92 \cdot 2^{(Qp-12)/6}. \tag{3.2}$$

After motion estimation, the coding mode, `mode`, for the MB that leads to the lowest R-D cost is selected by minimizing the following cost function:

$$J(\texttt{mode}, \lambda_{\texttt{mod}}) = SSD + \lambda_{\texttt{mod}} R(\texttt{mode}), \tag{3.3}$$

where SSD is the *sum of squared differences*, $R(\texttt{mode})$ is the number of bits needed to code the MB using `mode` and

$$\lambda_{\texttt{mod}} = \lambda^2_{\texttt{mot}} = 0.85 \cdot 2^{(Qp-12)/3}. \tag{3.4}$$

If $J_k(mode_k, \lambda_{mode_k})$ is the Lagrangian cost function of the $k^{th}$ MB that is coded with mode $mode_k$ and **mode** is the $N$-tuple $(mode_0, ..., mode_{N-1})$, where $N$ is the total number of MBs, a typical encoder assumes additive distortion and rate measures, resulting in the optimization problem:

$$\min_{\mathbf{mode}} \sum_{k=0}^{N-1} J_k(\mathbf{mode}, \lambda_{mode_k}) = \sum_{k=0}^{N-1} \min_{mode_k} J_k(mode_k, \lambda_{mode_k}). \tag{3.5}$$

That is, each MB selects the MB mode that gives the best R-D performance. In Section 3.2, we will describe how we optimize a subset of MBs concurrently, channeling resources to MBs that display the worst R-D performance.

Due to the inter-dependency between macroblocks (see Fig. 3.1), the R-D process of the current MB requires that the R-D process of its neighboring macroblocks be completed. To overcome the problem of excessive data dependency that is present within a frame, the proposed complexity scalable scheme uses the concept of *wavefront* [62] (see Fig. 3.2) to R-D optimize independent macroblocks in sets.

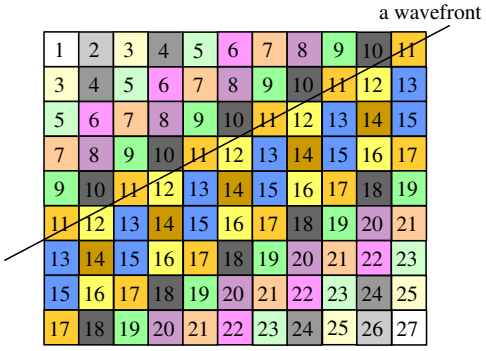



FIGURE 3.2: Wavefront MB scheduling: MBs on the same wavefront can be processed concurrently.

TABLE 3.1: Mode Mapping Functions

| Mode Mapping for ModeTest$(\cdot)$ | | | | | |
|---|---|---|---|---|---|
| input | SKIP/ INTRA | $16 \times 16$ | 16x8 | 8x16 | 8x8 |
| output | $16 \times 16$ | 16x8 | 8x16 | 8x8 | n.a. |
| | | | | | |
| Mode Mapping for ELModeTest$(\cdot)$ | | | | | |
| input | SKIP/ INTRA | $16 \times 16$ | 16x8 | 8x16 | 8x8 |
| output | $16 \times 16$ | $16 \times 16$ | 16x8 | 8x16 | 8x8 |

## 3.2 Complexity Scalable RDO

This section proposes a rate-distortion optimization (RDO) that allows the R-D computation of the video encoding process to be performed in a complexity scalable manner.

Let $B_{i,j}$ be the $i^{th}$ MB in the wavefront $W_j$. Each wavefront $W_j$ is processed in an ascending order $j$ to satisfy the dependencies between macroblocks. For each $W_j$ , all macroblocks are initially coded with the `SKIP` and `INTRA` mode, i.e.,

$$\text{Mode}(B) = \underset{m \in \{\texttt{INTRA},\texttt{SKIP}\}}{\arg\min} J(m, \lambda_{\texttt{mod}}), \quad \forall B \in W_j, \tag{3.6}$$

where Mode$(B)$ denotes the coding mode that gives the minimum R-D cost for MB $B$. Next, the encoder iteratively selects the MB, $B^* \in W_j$, with the highest cost to compute the R-D cost.

$$B^* = \underset{B \in W_j}{\arg\max}\, J(\text{Mode}(B), \lambda_{\texttt{mod}}). \tag{3.7}$$

The MB mode tested for $B^*$ is dependent on its previously tested mode (Table 3.1). Only one MB mode will be tested and we consider this one R-D operation. Let $\texttt{mode}_0 = \text{Mode}(B^*)$ and ModeTest$(\texttt{mode}_0)$ be the mode to be tested (see Table 3.1) given $\texttt{mode}_0$ ; then, if

$$J(\text{ModeTest}(\texttt{mode}_0), \lambda_{\texttt{mod}}) < J(\texttt{mode}_0, \lambda_{\texttt{mod}}), \tag{3.8}$$

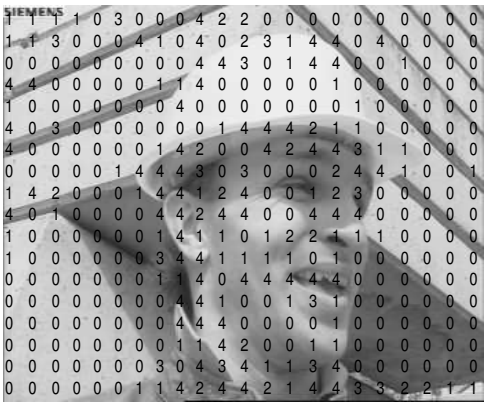



FIGURE 3.3: MBs in more complex regions tend to have higher priority (numbers show times MB is picked).

the tested MB $B^*$ is updated with its new coding mode,

$$\text{Mode}(B^*) = \text{ModeTest}(\texttt{mode}_0). \tag{3.9}$$

The iterative process, from eqn. (3.7) to (3.9), of selecting the next MB to compute its R-D cost continues until a predetermined number $N_{\text{op}}$ of R-D operations for the wavefront $W_j$ is done. This is given by

$$N_{\text{op}}(W_j) = \lfloor 2\beta \cdot |W_j| \rfloor, \quad \beta > 0, \tag{3.10}$$

where $|W_j|$ denotes the number of macroblocks in $W_j$ and $\beta$ is a control parameter. The motivation for the MB selection strategy in eqn. (3.7) is to divert computational resource to macroblocks with the worst R-D performance during the R-D optimization of a wavefront. Since a typical wavefront spans a large area across the image, it is likely to cover both areas with high and low motion activities. MBs in the more complex regions of the image tend to have higher priority (Fig. 3.3) during the selection, thus benefiting from the extra R-D operations. The performance of the algorithm can be improved if the MB that will benefit from the extra computation during each iteration can be better predicted. We observed that computing the 16x16 mode test often led to large R-D cost drop for the MB, so we modify the algorithm to give priority to MBs that have not gone through the 16x16 mode test. That is, though the same R-D cost comparison takes place before each iteration, all MBs along the wavefront has to be tested with the 16x16 mode before any MB is picked and tested with finer partition type. In the result section, we label this method, "method B", while the method that picks MBs based only on R-D cost comparison is labelled "method A". When $\beta$ is zero, the number of operation is zero; all MBs in the wavefront are coded as either SKIP or INTRA. When $\beta$ is 2, all MBs get 4 computations (all sub-MB types are tested during the 8x8 test); each MB would have tested all available modes. So, as $\beta$ varies from 0 to 2, the complexity of the encoder varies from one that uses only SKIP and INTRA mode to that of an exhaustive encoder.

**Algorithm 2**: Complexity Scalable RDO

**Data**: ModeTest(mode), (Table 3.1)
**Data**: ELModeTest(mode), (Table 3.1)
**Data**: *el*(boolean), frame is in enhancement layer.
**Data**: $N_{\text{wav}}$, number of wavefronts in the frame.
**Data**: $W_j$, wavefront number $j$.
**Data**: $N_{\text{op}}$, max. number of RDO iterations.
**for** $j \leftarrow 0$ ***to*** $N_{wav} - 1$ **do**
  **foreach** *MB* $B \in W_j$ **do**
    **if** *el* **then**
      $\text{Mode}(B) = \arg\min_{m \in \{\texttt{INTRA,SKIP,BL\_SKIP}\}} J(m, \lambda_{\texttt{mod}})$;
    **else**
      $\text{Mode}(B) = \arg\min_{m \in \{\texttt{INTRA,SKIP}\}} J(m, \lambda_{\texttt{mod}})$;
    **end**
  **end**
  $k \leftarrow 0$;
  $B^* = \arg\max_{B \in W_j} J(\text{Mode}(B), \lambda_{\texttt{mod}})$;
  **repeat**
    $prev\_mode \leftarrow \text{PrevMode}(B^*)$;
    **if** *el* **then**
      **if** $(prev\_mode == INTRA|SKIP|BL\_SKIP)$ **then**
        $test\_mode \leftarrow \text{ELModeTest}(Mode(Base(B^*)))$;
      **else**
        $test\_mode \leftarrow \text{ModeTest}(prev\_mode)$;
      **end**
    **else**
      $test\_mode \leftarrow \text{ModeTest}(prev\_mode)$;
    **end**
    **if** $J(test\_mode, \lambda_{\texttt{mod}}) < J(prev\_mode, \lambda_{\texttt{mod}})$ **then**
      $\text{Mode}(B^*) = test\_mode$;
    **end**
    $B^* = \arg\max_{B \in W_j | \text{PrevMode}(B) \neq 8\times 8} J(\text{Mode}(B), \lambda_{\texttt{mod}})$;
    $k \leftarrow k + 1$;
  **until** $(B^* == \emptyset) \| (k \geq N_{op}(W_j))$;
**end**

## 3.3 Complexity Control for SVC Enhancement Layer Refinement

As mentioned in the previous chapter, *Scalable Video Coding* (SVC), an extension of the H.264/AVC standard [63][64] provides efficient support for spatial, temporal and quality scalability. Though video scalability techniques have been proposed in the past, such as the scalable profiles for MPEG-2, H.263, and MPEG-4 Visual, they are less efficient and more complex than the SVC. SVC compresses video into a *base layer* that is H264/AVC encoded, and *enhancement layers* that provide additional information to scale the base or preceding layer in quality, spatial or temporal resolution.

In SVC, macroblocks in the enhancement layers have additional coding modes that are not available in the H.264/AVC. These additional coding modes are related to the new inter-layer prediction mechanisms introduced to reduce data redundancy between different video layers. The new coding modes allow the use of motion, residual and partitioning information of the lower spatial layers for prediction of the enhancement layer pictures:

- *Inter-Layer Intra Prediction*: When the corresponding block in the reference layer is intra-coded, the reconstructed data can be used as an intra prediction for the MB in the current layer.
- *Inter-Layer Motion Prediction* [65][66]: Data redundancy between motion vectors across layers can be reduced. An enhancement layer MB can re-use the motion and partitioning information (after appropriate scaling) from the co-located inter-coded block in the base layer. The `BL_SKIP` mode also allows an enhancement layer MB to inherit the motion information of its corresponding base layer MB.
- *Inter-Layer Residual Prediction* [65]: Inter-coded MB in the enhancement layer can, optionally with available inter-coding modes, utilize the upsampled residual information of the co-located block (intra or inter-coded) from the base layer as prediction. The prediction error of the residual information is coded in the enhancement layer.

Reusing base layer motion/partition/residual information may show better coding efficiency compared to coding the video layers separately. However, determining the optimal coding mode for each MB can be computationally intensive. Generally for each MB, the encoder has to exhaustively compute the R-D cost of all possible combination of coding modes (with and without inter-layer predictions) and partitions. It is therefore worthwhile to extend the aforementioned complexity scalable RDO scheme (see Section 3.2) to the enhancement layers.

For each wavefront in the enhancement layers, all the macroblocks are initially computed for their `SKIP`, `INTRA` and `BL_SKIP` modes. Then, the mode tested for each selected MB follows the mapping by ELModeTest$(\cdot)$ in Table 3.1, where mode tested depends on the best mode for corresponding base layer MB. This ensures that the base layer MB is never more finely-partitioned than the corresponding enhancement layer MB, reducing the modes that have to be tested to a subset of all possible modes. The choice of the mapping in ELModeTest$(\cdot)$ is based on the observation that the enhancement layer MB is often more finely partitioned than the base-layer MB (from Chapter 2). Subsequently, the mode to be tested for each selected MB is mapped by ModeTest$(\cdot)$ as in Section 3.2. Algorithm 2 outlines the complexity scalable RDO scheme for both the base and enhancement layer. PrevMode$(B)$ denotes the last tested coding mode for MB $B$.

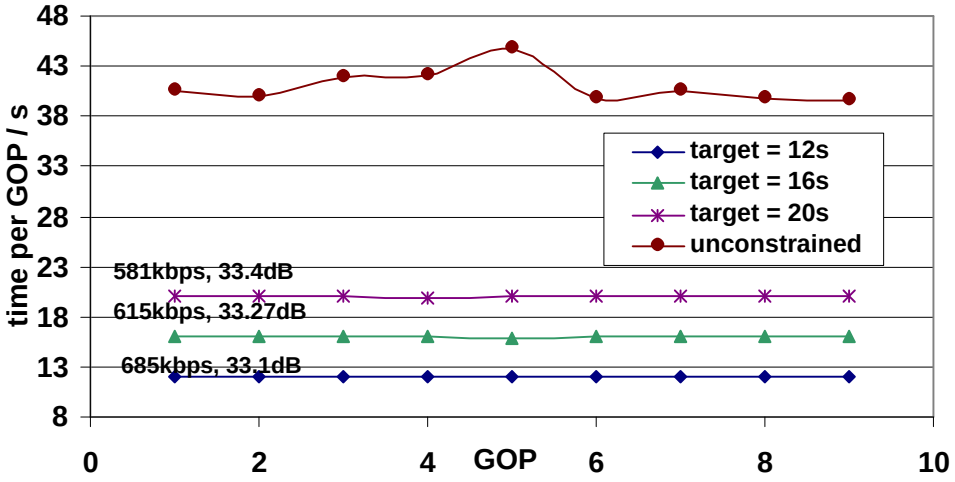



FIGURE 3.4: Adaptive Complexity control: $\beta$ is computed for each frame to maintain complexity of GOP encoding. Bit-rate and PSNR are shown for each complexity configuration. Result shown is for the sequence MOBILE.

## 3.4 Computational Resource Aware Encoding

In some applications where computational power is constrained or variable, it is necessary to adaptively adjust the encoder complexity to make best use of available resource. The video encoder can track the time taken to encode each frame and use the value as an indication of the level of available computational resource. Such an implementation will be useful for a software encoder that has no prior knowledge of the computational power of the computing platform it is running on. This design will also allow the video encoder to adapt to changing level of computing power (due to depleting batteries, new unrelated processes demanding computing cycles, new batteries, etc). We assumed that the encoding time of each frame varies linearly with $\beta$. That is, $T_f = \mu_1\beta + \mu_0$. We update the model parameters $\mu_0$ and $\mu_1$ after the encoding of each frame by observing the encoding time and the time required to code the SKIP and INTRA modes of the frame. This will help track sequence source characteristics and computing resource. $\beta$ for the next frame is then adjusted according to the updated model to meet complexity requirement (Algorithm 3).

As computers become more and more ubiquitous, computing platforms also become more diverse. If a designer has to implement software encoders on a few platforms, a different encoder may have to be designed if the computational power of the computing platforms differ. An attractive feature of the encoding scheme described in this section is that the same implementation of the algorithm can potentially work on different computing platforms, since the complexity controlling parameter, $\beta$ is adjusted dynamically and the available computing resource is apparent to the encoder through the time taken to encode each frame. The algorithm is also well-suited to adapt to changing source characteristics or varying computing resource.

**Algorithm 3**: Adaptive Complexity Control

**Input**: $T$, remaining encoding time for GOP .
**while** *(GOP*$\leftarrow$*Fetch_Next_GOP())*$\neq \emptyset$ **do**
  **Data**: $t_0$, encoding time (`SKIP` and `INTRA`) for FRAME.
  **Data**: $t_1$, encoding time for FRAME.
  **Data**: $(\mu_0, \mu_1)$, model parameters.
  **while** *(FRAME*$\leftarrow$*Fetch_Next_FRAME())*$\neq \emptyset$ **do**
    $(t_0, t_1) \leftarrow$`Encode(`*FRAME*, $\beta$`);`
    $\mu_0 \leftarrow t_0$;
    $\mu_1 \leftarrow (t_1 - t_0)/\beta$;
    $T \leftarrow T - t_1$;
    $\beta \leftarrow (T/$`NO_REMAINING_FRAME()` $- \mu_0)/\mu_1$;
  **end**
**end**

This signal processing-centric approach [67] to optimization given a computational constraint is similar to the approach in [68] which trades off compression performance with computation of a particular compression algorithm by adjusting parameters of the encoder. The proposed compression scheme has the added ability of dynamic adaptation to changing source signal and computing resource constraints.

Voltage scaling can reduce energy consumption by at the cost of slower circuits [69]. Using the described algorithm, the video encoder can respond to energy saving actions that result in lower computational power. Energy constraint devices can also make high level voltage scaling decisions knowing that the video encoder can adjust its complexity in response.

## 3.5 Experimental Results

The experiment uses a modified version of JSVM 8.10 with CIF test sequences at 15 fps. The GOP size is 16 and uses the hierarchical B-pictures coding structure [43][70]. The quantization parameter $Qp$ is varied to evaluate the effectiveness of the algorithm at different bit-rates. The search range was set to 96 and the maximum number of reference pictures in each reference list was 1.

### 3.5.1 H.264 encoder complexity control

To demonstrate the complexity scalable nature of the proposed encoding scheme, $\beta$ of the encoder is varied and the coding performance is recorded in Table 3.5 (sequences are coded with $Qp = 24, 30, 36, 42$). It is clear from Fig. 3.5 - Fig. 3.8 that the complexity-coding performance trade-off can be controlled with a single parameter $\beta$. Since setting $\beta = 1.0$ gives result close to what is possible with exhaustive R-D optimization, only results for $\beta$ up to 1.0 are shown.

TABLE 3.2: Complexity control parameter $\beta$ and Reduction of Function Calls

| | No of function calls ($\times 10^3$) | | | | |
|---|---|---|---|---|---|
| Function | `xGetSAD4x/ 8x / 16x` | | | `transform4x4Blk` | `xEncode4x4InterBlock` |
| FOREMAN | | | | | |
| *original* | 96307 | 56571 | 12708 | 2217 | 1379 |
| $\beta = 1$ | 45046 | 26600 | 7415 | 1759 | 919 |
| $\beta = 0.8$ | 32874 | 19594 | 5837 | 1661 | 816 |
| $\beta = 0.6$ | 21891 | 13162 | 4568 | 1580 | 726 |
| $\beta = 0.4$ | 10745 | 6665 | 3159 | 1501 | 631 |
| $\beta = 0.2$ | 1531 | 1104 | 1750 | 1447 | 546 |
| MOBILE | | | | | |
| *original* | 107000 | 60384 | 12997 | 2213 | 1381 |
| $\beta = 1$ | 46959 | 26627 | 7352 | 1748 | 914 |
| $\beta = 0.8$ | 32475 | 18757 | 5790 | 1644 | 808 |
| $\beta = 0.6$ | 21166 | 12377 | 4583 | 1563 | 722 |
| $\beta = 0.4$ | 9614 | 5893 | 3130 | 1482 | 630 |
| $\beta = 0.2$ | 1358 | 969 | 1697 | 1430 | 551 |
| CREW | | | | | |
| *original* | 93651 | 56031 | 12809 | 2230 | 1374 |
| $\beta = 1$ | 44198 | 26394 | 7377 | 1781 | 914 |
| $\beta = 0.8$ | 32154 | 19485 | 5732 | 1684 | 810 |
| $\beta = 0.6$ | 21583 | 13343 | 4558 | 1606 | 720 |
| $\beta = 0.4$ | 10495 | 6641 | 3108 | 1525 | 623 |
| $\beta = 0.2$ | 1524 | 1178 | 1770 | 1473 | 538 |
| SOCCER | | | | | |
| *original* | 97201 | 57432 | 13010 | 2217 | 1379 |
| $\beta = 1$ | 45629 | 27074 | 7514 | 1759 | 918 |
| $\beta = 0.8$ | 32936 | 19671 | 5948 | 1659 | 810 |
| $\beta = 0.6$ | 22113 | 13381 | 4737 | 1582 | 721 |
| $\beta = 0.4$ | 11552 | 7103 | 3232 | 1529 | 621 |
| $\beta = 0.2$ | 2455 | 1665 | 1796 | 1527 | 523 |

Motion estimation requires the functions `xGetSAD4x`, `xGetSAD8x` and `xGetSAD16x` to compute the SAD distortion; Functions `transform4x4Blk` and `xEncode4x4InterBlock` are called during R-D operations to encode a MB/ sub-block in each possible coding mode.

Much of the coding performance of the exhaustive encoder can be achieved (to within 0.1-0.27dB for GOP16 and 0.1-0.25dB for IPPP) when the complexity of the encoder is scaled back to achieve 42-50% complexity reduction. This complexity reduction performance is close to what is possible with algorithms specifically designed for complexity reduction(eg. [48] (IPPP): 0.02-0.2dB drop with 32-58% complexity reduction (table 3.3 compares the proposed algorithm against [48] for 3 sequences) and [34] (IPPP): 9-45% complexity reduction with negligible PSNR drop). Our scheme can also be easily extended to a GOP coding structure.

It is conceivable that the proposed complexity scalable encoding strategy can be implemented with other fast mode decision algorithms to achieve superior performance.

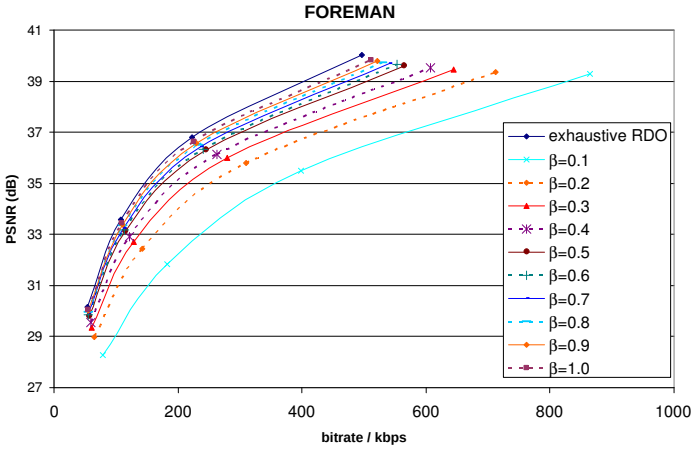



FIGURE 3.5: Complexity scalable encoding: rate-distortion performance against parameter $\beta$, with Method A.

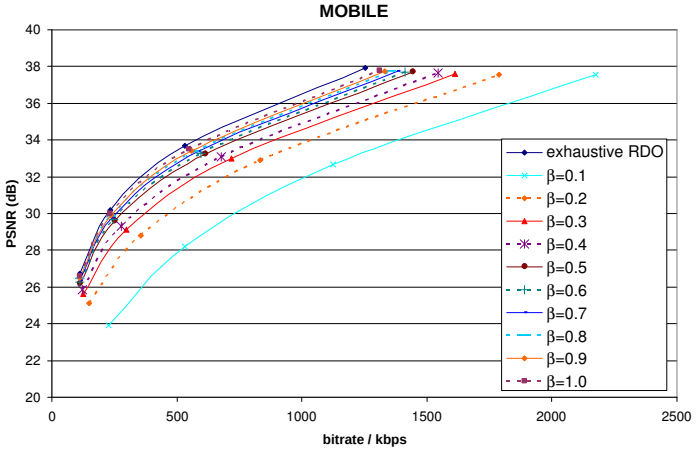



FIGURE 3.6: Complexity scalable encoding: rate-distortion performance against parameter $\beta$, with Method A.

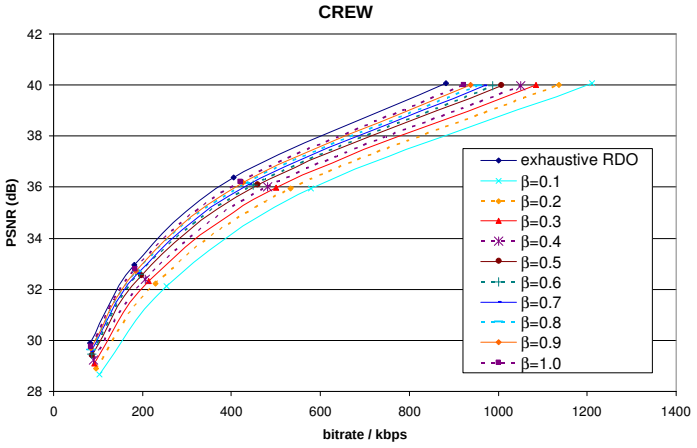



FIGURE 3.7: Complexity scalable encoding: rate-distortion performance against parameter $\beta$, with Method A.

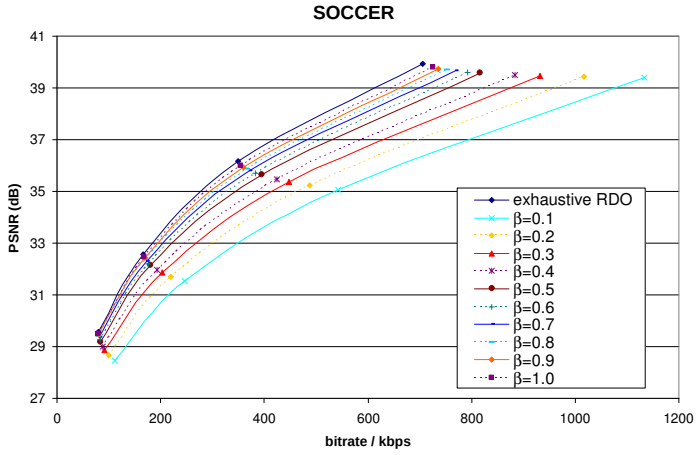



FIGURE 3.8: Complexity scalable encoding: rate-distortion performance against parameter $\beta$, with Method A.

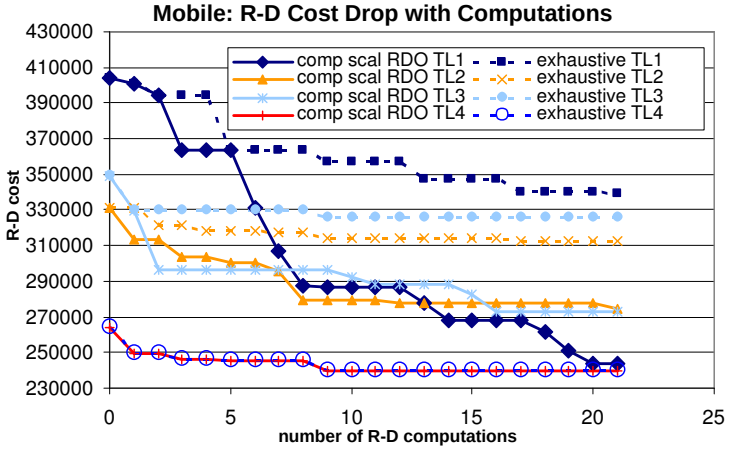



FIGURE 3.9: R-D cost drop with computations: Total R-D cost of all MBs in Wavefront number 22 of several CIF frames of different temporal levels. Total R-D cost drop significantly faster when MBs in a wavefront are optimized concurrently.

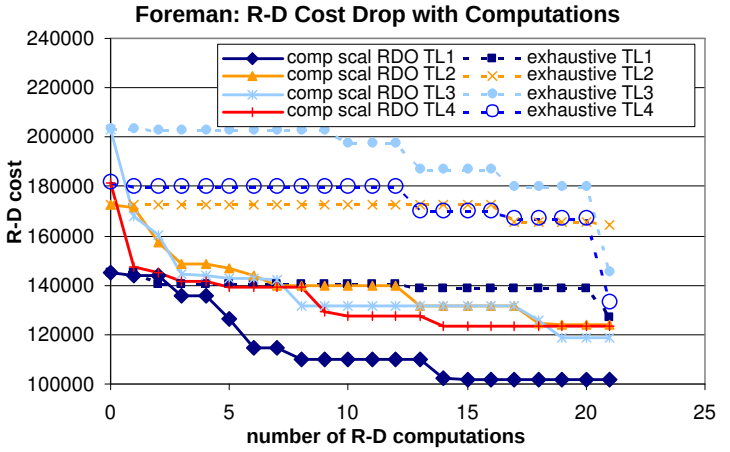



FIGURE 3.10: R-D cost drop with computations: Total R-D cost of all MBs in Wavefront number 22 of several CIF frames of different temporal levels. Total R-D cost drop significantly faster when MBs in a wavefront are optimized concurrently.

TABLE 3.3: Results - comparison of complexity reduction against Lin et al.

| | IPPP Method B with $\beta = 1$ | | IPPP [48] | |
|---|---|---|---|---|
| | speedup | BD-PSNR | speedup | BD-PSNR |
| foreman | 43.37 | -0.17 | 42.88 | -0.09 |
| football | 41.41 | -0.12 | 32.99 | -0.21 |
| bus | 43.66 | -0.22 | 35.76 | -0.06 |

TABLE 3.4: Results - Comparing Methods A and B (speedup shows percentage complexity reduction over an exhaustive encoder, computed with average encoding time over 4 values of $Qp$, BD-PSNR computed against an exhaustive encoder.)

| | | GOP16 (Method A) | | GOP16 (Method B) | |
|---|---|---|---|---|---|
| | $\beta$ | speedup | BD-PSNR | speedup | BD-PSNR |
| crew | 0.2 | 83.55 | -1.65 | 88.53 | -1.37 |
| | 0.4 | 74.10 | -1.08 | 83.61 | -0.62 |
| | 0.6 | 63.7 | -0.67 | 76.98 | -0.36 |
| | 0.8 | 54.34 | -0.45 | 63.38 | -0.22 |
| | 1.0 | 43.39 | -0.27 | 50.03 | -0.12 |
| foreman | 0.2 | 81.52 | -2.33 | 87.00 | -2.15 |
| | 0.4 | 71.45 | -1.25 | 83.61 | -0.62 |
| | 0.6 | 60.43 | -0.68 | 76.41 | -0.55 |
| | 0.8 | 50.78 | -0.42 | 60.73 | -0.32 |
| | 1.0 | 40.19 | -0.21 | 47.50 | -0.15 |
| soccer | 0.2 | 79.37 | -2.31 | 87.00 | -2.15 |
| | 0.4 | 68.00 | -1.41 | 82.34 | -1.01 |
| | 0.6 | 56.7 | -0.76 | 76.41 | -0.55 |
| | 0.8 | 47.06 | -0.42 | 60.73 | -0.32 |
| | 1.0 | 36.84 | -0.20 | 47.50 | -0.15 |

Figs. 3.9 and 3.10 show the drop in the R-D cost of a wavefront as the number of R-D computations increases. To show the effectiveness of the complexity scalable algorithm at quickly reducing the total R-D cost of a wavefront, the change in total R-D cost of a particular wavefront is tracked and compared to a wavefront that is being optimized MB-wise (each MB is optimized by testing all MB modes before moving to the next MB in the wavefront). Since computation is channeled to MB with high R-D cost, the algorithm decreases the total R-D cost of the wavefront significantly faster.

The algorithm's ability to decrease the total R-D cost of a wavefront quickly improves the performance of the encoder in 2 ways:

- The proposed encoding scheme can attain the coding performance near to that of an exhaustive search encoder with a significantly smaller number of computation.
- By controlling the number of computations per wavefront, the complexity of the encoder can be controlled. The computational resource will always be channeled to the curent worst performing (in a R-D cost sense) MB.

TABLE 3.5: Results - H.264 complexity control (speedup shows percentage complexity reduction over an exhaustive encoder, computed with average encoding time over 4 values of $Qp$, BD-PSNR computed against an exhaustive encoder.)

| | | GOP16 (Method B) | | IPPP (Method B) | |
|---|---|---|---|---|---|
| | $\beta$ | speedup | BD-PSNR | speedup | BD-PSNR |
| crew | 0.2 | 88.53 | -1.37 | 84.22 | -0.96 |
| | 0.4 | 83.61 | -0.62 | 79.06 | -0.46 |
| | 0.6 | 76.98 | -0.36 | 72.89 | -0.30 |
| | 0.8 | 63.38 | -0.22 | 59.38 | -0.19 |
| | 1.0 | 50.03 | -0.12 | 46.42 | -0.10 |
| foreman | 0.2 | 87.00 | -2.15 | 76.91 | -0.81 |
| | 0.4 | 82.34 | -1.01 | 72.23 | -0.97 |
| | 0.6 | 76.41 | -0.55 | 65.68 | -0.53 |
| | 0.8 | 60.73 | -0.32 | 54.13 | -0.31 |
| | 1.0 | 47.50 | -0.15 | 43.37 | -0.17 |
| soccer | 0.2 | 87.00 | -2.15 | 76.91 | -1.81 |
| | 0.4 | 82.34 | -1.01 | 72.23 | -0.97 |
| | 0.6 | 76.41 | -0.55 | 65.68 | -0.53 |
| | 0.8 | 60.73 | -0.32 | 54.13 | -0.31 |
| | 1.0 | 47.50 | -0.15 | 43.37 | -0.17 |
| football | 0.2 | 88.37 | -1.29 | 83.40 | -0.87 |
| | 0.4 | 82.77 | -0.66 | 77.67 | -0.44 |
| | 0.6 | 77.06 | -0.39 | 72.23 | -0.29 |
| | 0.8 | 61.57 | -0.27 | 55.82 | -0.19 |
| | 1.0 | 47.48 | -0.16 | 41.41 | -0.12 |
| city | 0.2 | 85.85 | -2.94 | 71.43 | -2.15 |
| | 0.4 | 82.31 | -1.23 | 68.61 | -0.96 |
| | 0.6 | 77.59 | -0.67 | 64.07 | -0.52 |
| | 0.8 | 62.62 | -0.35 | 53.25 | -0.20 |
| | 1.0 | 50.71 | -0.22 | 43.72 | -0.10 |
| bus | 0.2 | 86.63 | -2.95 | 78.47 | -2.70 |
| | 0.4 | 82.92 | -1.53 | 74.04 | -1.36 |
| | 0.6 | 78.04 | -0.88 | 69.02 | -0.83 |
| | 0.8 | 63.50 | -0.50 | 55.46 | -0.43 |
| | 1.0 | 50.12 | -0.27 | 43.66 | -0.22 |
| silent | 0.2 | 83.99 | -0.67 | 72.44 | -0.59 |
| | 0.4 | 80.74 | -0.52 | 67.56 | -0.48 |
| | 0.6 | 75.64 | -0.33 | 64.44 | -0.31 |
| | 0.8 | 57.08 | -0.12 | 51.11 | -0.12 |
| | 1.0 | 45.04 | -0.04 | 40.89 | -0.05 |

Other than demonstrating the reduction in the number of function calls that were made during the encoding process, table 3.2 shows, not surprisingly, that the reduction is across all MB operations at the MB level. The proposed complexity scalable scheme can be useful as it can be expected to work well with other complexity reduction techniques.

A scheme that tries to control encoder complexity by controlling the motion estimation search

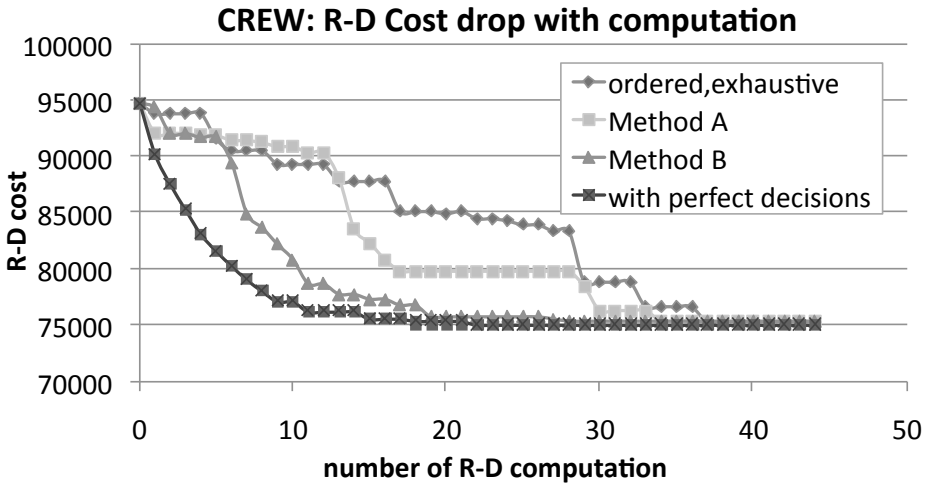



FIGURE 3.11: R-D cost drop with computations: Total R-D cost of all MBs in a wavefront. Total R-D cost drops faster when MBs in a wavefront are optimized concurrently. The R-D cost of Method B drops at a rate close to what is achievable with perfect decisions.

range will become less effective if a faster implementation of the SAD operation is used (possibly through the use of SIMD instructions or some effective fast search algorithms) and the SAD operations are no longer the bottleneck in the encoding operations. Controlling the computational resource allocation as proposed tends to channel limited resource to MBs that benefit most from the extra computations. Since a R-D computation for a particular partition includes different operations that can be computationally complex (eg. motion estimation, transforms and inverse transforms for the computation of rates and distortions), good complexity control can be expected even if these sub-operations of the R-D computation are implemented with lower complexity.

Fig. 3.11 shows the drop in the R-D cost of a wavefront as the number of R-D computations increases. To show the effectiveness of the complexity scalable algorithm at quickly reducing the total R-D cost of a wavefront, the change in total R-D cost of a particular wavefront is tracked and compared to a wavefront that is being optimized MB-wise (each MB is optimized by testing all MB modes before moving to the next MB in the wavefront). Since computation is channeled to MB with higher R-D cost, the algorithm decreases the total R-D cost of the wavefront significantly faster. The R-D cost changes when the wavefront is coded with prior knowledge of the R-D costs associated with all the modes for each MB is also shown. In this case, the encoder can always choose to operate on the MBs for which the next operation will bring the largest R-D cost drop, resulting in perfect decision making. The plots shows that with the proposed method, the R-D cost drops at a rate much closer to encoder with perfect decision making.

The algorithm described in [57] controls the complexity of the encoder by controlling the number of SKIP MBs within a video frame. For comparison, we implemented a H.264 compliant (skip-or-code) encoder that captures the essence of the technique. The key to their

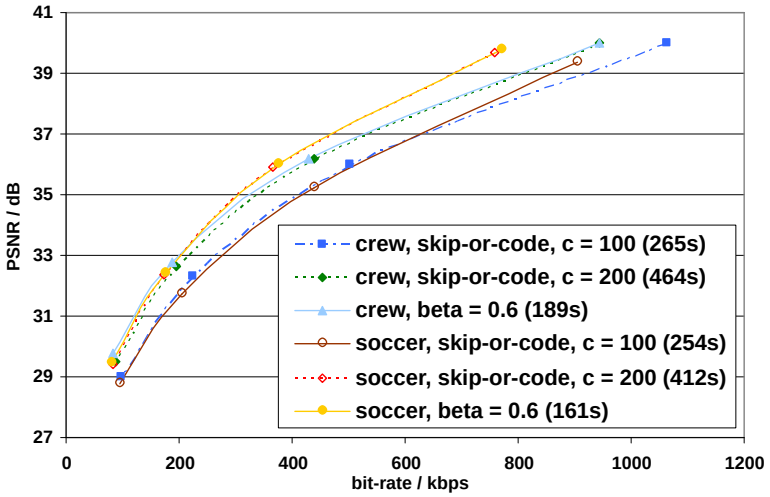



FIGURE 3.12: Comparing proposed scheme with a scheme that controls complexity through the proportion of SKIP MBs. c is the number of MBs coded. Number shown is the encoding time.

method lies in being able to decide before coding a MB whether the MB should be skipped or coded. The proportion of MBs coded will determine the encoding complexity of the frame. In our experiment, we assumed that, when making skip/code decisions, the skip-or-code encoder always chooses to code MBs that will experience the most R-D performance improvement after coding. That is, when a specified number of MBs within a frame are coded, the encoder will always choose the MBs with the largest RD(SKIP/INTRA) - RD(code), where RD(SKIP/INTRA) is the R-D cost of the MB when coded with the SKIP/INTRA mode and RD(cost) is the R-D cost of the MB if it is coded. To achieve this, we ran the H.264 encoder with exhaustive mode decisions to collect the R-D costs of each MB when coded with different modes. The data collected was then used to assist the skip-or-code encoder during encoding. Fig. 3.12 shows that even if we assume perfect skip/code decisions by the encoder (that is, the encoder will always pick the correct MBs to code), the skip-or-code encoding scheme still underperforms our proposed algorithm. This highlights the value of optimizing a set of MBs concurrently.

### 3.5.2 SVC encoder complexity control

The effectiveness of the complexity control algorithm can also be observed in the scalable extension. In the experiment, $Qp$ and $\beta_0$ (which controls the number of R-D computation per MB per wavefront in the base layer) of the base layer is fixed while $Qp$ of the enhancement layer and $\beta_1$ are varied to obtain the R-D curves.

From the R-D curves (Fig. 3.13 - 3.16), it is clear that the encoding time on the same computing platform (or equivalently the power consumption of the video encoder) can be controlled by a single parameter.

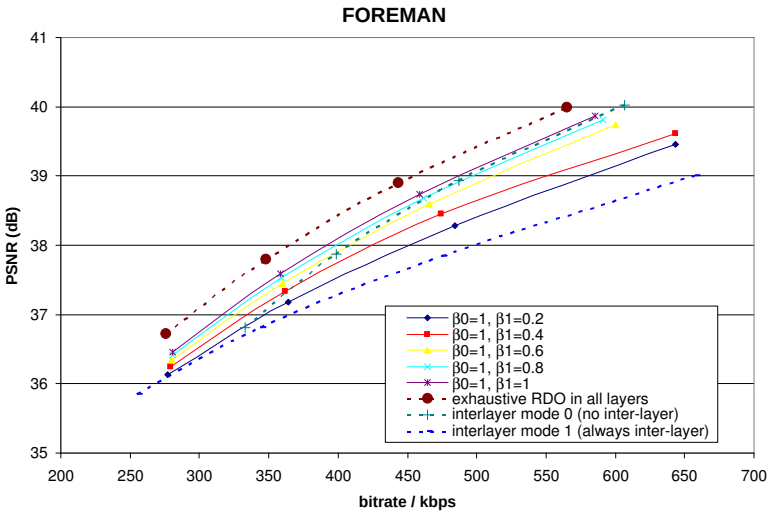



FIGURE 3.13: Complexity scalable encoding: rate-distortion performance (2 CGS layers (CIF)) against parameter $\beta$. $\beta_0$ is fixed at 1 while $\beta_1$ is varied. Base layer is encoded with $Qp$ set at 36 in the config. file. Enhancement layer $Qp$ is varied (30, 28, 26, 24) to obtain the R-D curves.

The ability to control the complexity of the encoding at each layer also provides insights on the allocation of computational resource across the layers. From Fig.3.13, when interlayer mode is 1 (the enhancement layer always reuses the base layer motion information), coding performance at the enhancement layer suffers as the motion information acquired at the base layer is not optimized for the enhancement layer. Encoding the enhancement layer in this mode is however, relatively less complex as no R-D computation is carried out in the enhancement layers.

The curve 'interlayer mode 1' in Fig.3.13 shows the performance of the reference encoder when all computational resource is invested in acquiring an optimal motion vector field for the base layer. The enhancement layer then reuses this information and refines only the residual information. Setting $\beta_0$ to 1 and $\beta_1$ to 0.2 results in an encoder of lower complexity but superior coding performance. Since result in the previous sub-section shows that setting $\beta_0$ to 1 has little effect on the performance of single layer coding, we can conclude that such an allocation brings about better overall coding performance.

By setting $\beta$ values independently in different layers, the computational resource allocation to each layer can be controlled leading to a more efficient use of resource. As has been shown in the experiment, channeling resource from base to enhancement layer is likely to lead to better performance compared to only optimizing motion information in the base layer.

Optimizing the base layer and then reusing the motion information in the enhancement layers is a possible low complexity option provided by the specification. However, when there is a constraint in computational resource, the experiment results show that it is not the best way to allocate limited resource and investing some resource in the refinement of motion information in the enhancement layers will probably lead to better overall coding performance.

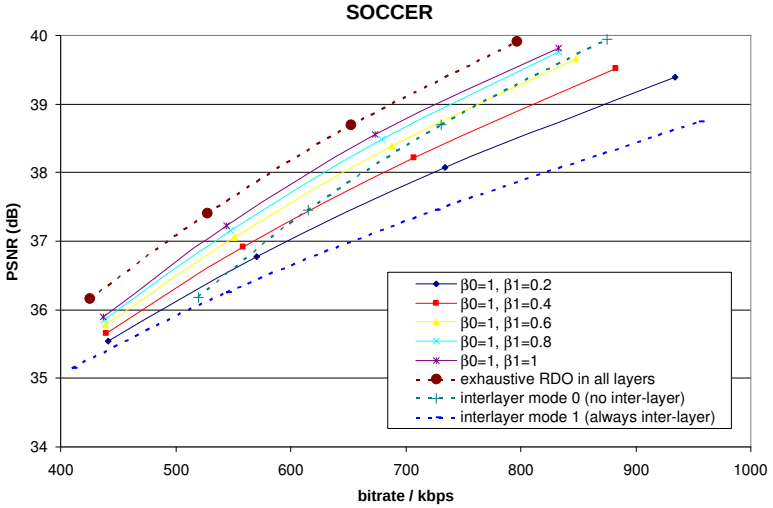



FIGURE 3.14: Complexity scalable encoding: rate-distortion performance (2 CGS layers (CIF)) against parameter $\beta$. $\beta_0$ is fixed at 1 while $\beta_1$ is varied. Base layer is encoded with $Qp$ set at 36 in the config. file. Enhancement layer $Qp$ is varied (30, 28, 26, 24) to obtain the R-D curves.

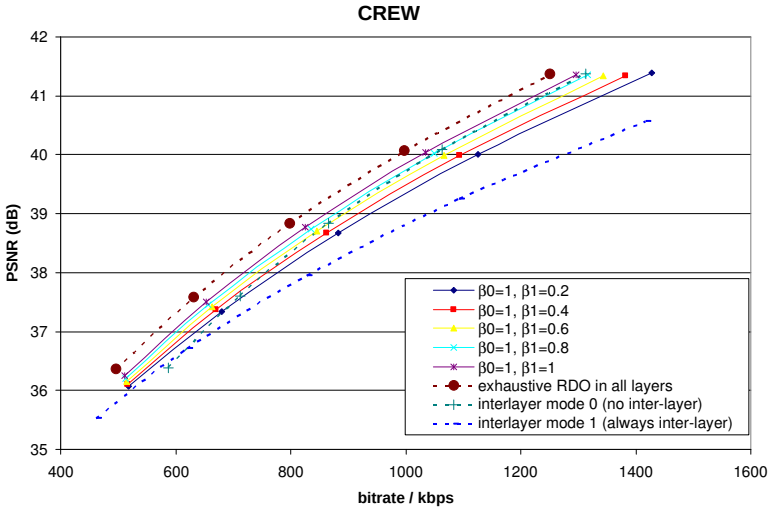



FIGURE 3.15: Complexity scalable encoding: rate-distortion performance (2 CGS layers (CIF)) against parameter $\beta$. $\beta_0$ is fixed at 1 while $\beta_1$ is varied. Base layer is encoded with $Qp$ set at 36 in the config. file. Enhancement layer $Qp$ is varied (30, 28, 26, 24) to obtain the R-D curves.

Figs. 3.17 and 3.18 show a series of R-D curves that show the performance of the encoder with different allocation of computational resource between the base and one enhancement layer. The curves show the PSNR of sequences when both layers are decoded.

When less resource was invested during the rate-distortion optimization in the base layer, the coding efficiency at the enhancement layer can also be affected. When the base layer motion information is closer to optimal, due to the higher number of computation that was carried out in the complexity scalable scheme, the base layer motion information available for reuse also appeared to be better suited for the enhancement layer (despite it being optimized for a lower bit-rate).

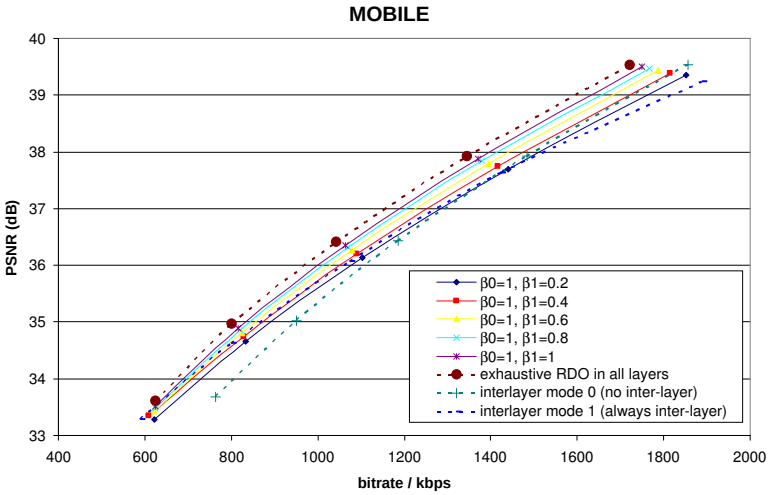



FIGURE 3.16: Complexity scalable encoding: rate-distortion performance (2 CGS layers (CIF)) against parameter $\beta$. $\beta_0$ is fixed at 1 while $\beta_1$ is varied. Base layer is encoded with $Qp$ set at 36 in the config. file. Enhancement layer $Qp$ is varied (30, 28, 26, 24, 22) to obtain the R-D curves.

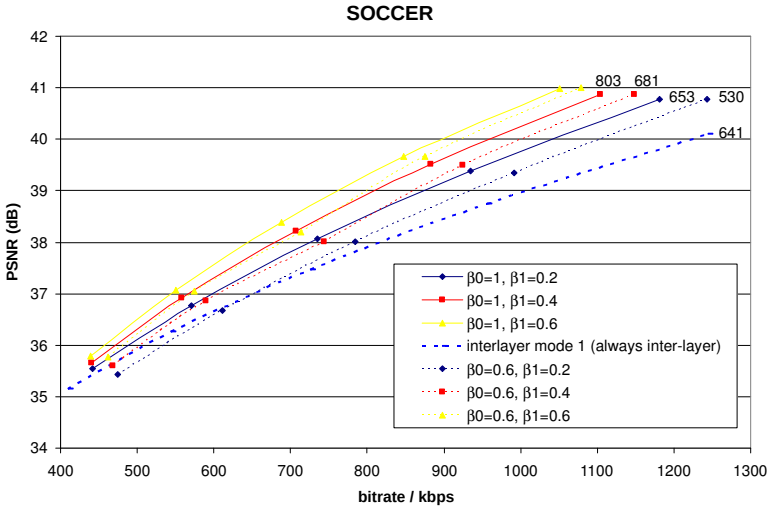



FIGURE 3.17: Complexity scalable encoding: rate-distortion performance (2 CGS layers (CIF)) against parameter $\beta$. $\beta_0$ is fixed at 0.6 and 1 while $\beta_1$ is varied. Base layer is encoded with $Qp$ set at 36 in the config. file. Enhancement layer $Qp$ is varied (30, 28, 26, 24, 22) to obtain the R-D curves. Number shows the encoding time for a sequence of 150 frames (an indication of encoder complexity).

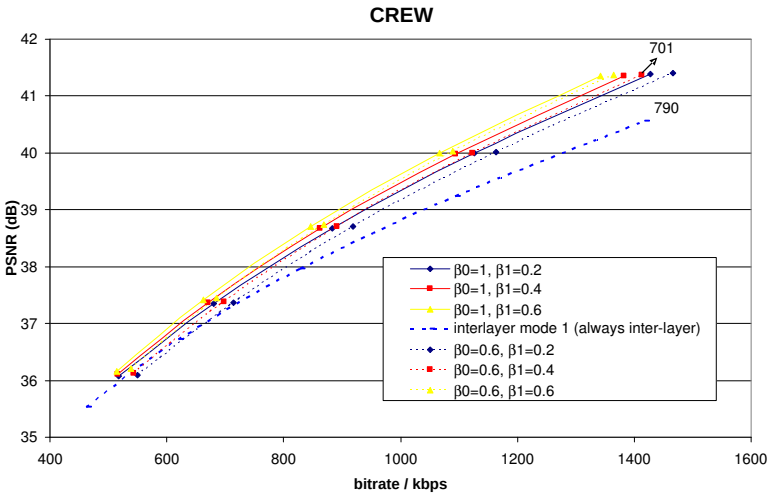



FIGURE 3.18: Complexity scalable encoding: rate-distortion performance (2 CGS layers (CIF)) against parameter $\beta$. $\beta_0$ is fixed at 0.6 and 1 while $\beta_1$ is varied. Base layer is encoded with $Qp$ set at 36 in the config. file. Enhancement layer $Qp$ is varied (30, 28, 26, 24, 22) to obtain the R-D curves. Number shows the encoding time for a sequence of 150 frames (an indication of encoder complexity).

As the bit-rate between the base layer and the enhancement layer widens, the motion vector information becomes less optimal for the enhancement layer and the coding performance worsens relative to when exhaustive R-D optimization is used.

Being able to separately control the complexity of encoding at each layer allows the encoder to optimize each layer to different extent depending on the importance of each layer or clients' requirements. When computational resource is limited, channeling resource to the R-D operations in the enhancement layers at the expense of base layer may be a good idea, especially when the bit-rates of the two layers are significantly different and the base layer motion information is far from optimized for the enhancement layers.

## 3.6 Remarks

In this chapter, we introduce a complexity scalable encoding scheme suitable for H.264/AVC. With a complexity scalable framework, encoding complexity can be reduced with a graceful degradation in coding performance to meet power constraints or real-time encoding requirements. During scalable video encoding, the encoding complexity of each layer can be controlled independently, making the allocation of computational resource across layers possible.

The heterogeneity of computing platforms means that computation-intensive algorithms (eg. video compression) must be optimized in a machine dependent manner. The encoding framework introduced in this chapter enables the design of source signal and computational resource adaptive video encoding applications that make full use of computing resource while operating under time constraints.

In chapters that follow, we will argue the importance of complexity control, both as a useful to help encoders operating under computational resource contraints and as essential mechanism to allow flexible resource allocation.

# Chapter 4

# Rate Control

## 4.1 Introduction

H.264 rate control is a difficult problem that has motivated numerous possible solutions. The challenge lies in determining a quantization parameter ($Qp$) that will be used for both the rate-distortion (R-D) optimization process and the quantization of transform coefficients. In this work, we attempt to achieve effective rate control with a different approach. By modelling the relationships of distortion, texture bits, non-texture bits and $Qp$, we can derive the $Qp$ required for both R-D optimization and quantization through Lagrangian optimization. From experiments with several video sequences, we found that our rate control scheme is capable of effective rate control with minimal model updates during encoding. The proposed rate control scheme adapts quickly to the characteristic of the source data and is particularly effective at controlling the rate of videos with high and unpredictable motion content.

Controlling the bit-rate of video is important for ensuring effective channel adaptation during its delivery. When transporting video for real-time communication under bit-rate constraint, video data that cannot be sent in time will be accumulated in the encoder buffer. If the number of encoding bits used fluctuates greatly from frame to frame, a large buffer is required. Also, when video frames cannot arrive on time at the decoder, continuous play-back is not possible. This inevitably affects the performance of low-delay communication systems.

Rate control and adaptation is also important during wireless transmission under limited radio resources. While power control maintains the capacity of the transmission link between source and destination, rate control ensures that the source rate is appropriately constrained according to channel conditions. Rate control algorithms that are capable of producing video bit-streams of constant bit-rate in spite of varying source characteristics are also useful during provisioning of large scale multimedia communication system where services have to be provided to a large number of users.

Although rate control algorithms are not specified by video coding standards, different standards, each with distinct characteristics, require different solutions. Some examples of these are TM5 for MPEG-2[71], TMN8 for H.263[72] and VM8 for MPEG-4[73]. The specification of the quantization scheme is especially germane to the design of rate control algorithms as it determines the rate and distortion characteristics of the encoder. These algorithms achieve rate control by modelling the relation between rate, $R$ and quantization parameter, $Qp$. TM5 uses a simple model: $R(Qp) = X/Qp$, where $X$ is a constant. Subsequent rate control schemes use more accurate and complex quadratic rate distortion models [74]. TMN8 models the rate-quantization relationship of a MB by $R = K\sigma^2/Qp^2 + C$, where $K$ and $C$ are constants and $\sigma^2$ is the variance of prediction residues in the MB. VM8 takes the mean absolute difference (MAD) between predicted and reference macroblocks into consideration, using the model: $R = X_1\mathtt{MAD}/Qp + X_2\mathtt{MAD}/Qp^2$, where $X_1$ and $X_2$ are model parameters.

Based on a simple rate model, $\rho$-domain rate control algorithms are proposed for different DCT-based coding schemes in [75]; $\rho$ indicates the proportion of zero coefficients after quantization. Another possible approach is to encode each video unit several times before selecting the best parameters[76][77]. Though simple, such methods may not be suitable for applications that operate under power or computational resource constraints.

H.264/AVC[35] is a high performance video coding standard developed by the Joint Video Team (JVT) that is expected to be widely adopted. H.264 rate control is a challenging problem. Due to the presence of many coding tools that help improve prediction performance of each MB, the proportion of non-texture header bits in the encoded file is generally higher compared to that in previous standards. For video sequences with complex or varying motion characteristics, the proportion and the number of non-texture bits are also difficult to predict. The final bit-rate of an encoded video is dependent on mode decisions, choice of motion vectors and the extent to which the transformed coefficients are quantized. If header bits do not make up a significant portion of the encoded video file, it is conceivable that a reasonably well-performing rate control scheme can be designed by only considering texture bits. In H.264, however, the header bits required to code mode decisions information and the increased number of motion vectors make it necessary to consider the impact of non-texture bits during the design of an effective rate control scheme.

Different coding modes show different coding performance at different rates. A typical encoder that carries out rate-distortion (R-D) optimization weighs the distortion, $D$ against the rate, $R$ of every mode and chooses the mode that results in the lowest R-D cost for each MB. The Lagrangian formulation of the R-D optimization is: min $J$, where $J = D + \lambda R$. The work in [60] demonstrated that using a $Qp$-dependent $\lambda$ is important for achieving good coding performance over a range of bit-rates. This poses a problem for rate control schemes that derive quantization parameters by examining the MAD or variance of prediction residue: the prediction residue is available only after the best coding mode is chosen through R-D optimization but the $Qp$ is required for the optimization process.

In [78], rate control is achieved by controlling the $Qp$ of each basic unit, which can be a frame, a group of MBs or a single MB. To determine the $Qp$ at the start of the R-D optimization, a predicted MAD is used. This value is predicted from the MAD of previously encoded basic units using a linear model that is updated after the encoding of each basic unit through linear regression. Several subsequent schemes aim to improve rate control performance by improving the accuracy of MAD prediction. Since each basic unit used during regression can be coded with different modes with different texture-non-texture bits balance, predicting the MAD can be both unwieldy and inaccurate. Even if the MAD prediction and the MAD-dependent rate-quantization relationship update are reliable, the unpredictable nature of the number of non-texture may lead to inaccurate rate control.

Recognizing the need to predict non-texture bits more accurately, Kwon et al[79] proposed a prediction model that relates the number of non-zero motion vectors to the number of non-texture bits. Using the model, the number of non-texture bits is predicted after R-D optimization, the $Qp$ for quantization is then determined based on the remaining bit budget for the encoded MB. This $Qp$ can be different from the $Qp$ that was used for R-D optimization; only the quantization $Qp$ is coded as the $\lambda$ determining $Qp$ used during R-D optimization is an encoder-only issue that need not be communicated to the decoder.

In this work, we attempt to achieve rate control with a different approach. In our rate control scheme, the $Qp$ used for R-D optimization, $Qp_m$ can be different from the $Qp$ used to determine the quantizer step-size, $Qp_t$. We first model the relationships of distortion $D$, texture bits, $R_t$, non-texture bits, $R_m$ and the two $Qp$s. With these models, we derive the values of $Qp_m$ and $Qp_t$ through Lagrangian optimization. From experiments on several sequences, we found that our proposed scheme is capable of good rate control performance with some simple updates of the models after the encoding of each basic unit. Our rate control scheme adapts quickly to the characteristic of the source data and is particularly effective at controlling the rate of videos with high and unpredictable motion content.

The rest of the chapter is organized as follows. In Section 4.2, we discuss the difficulty of implementing both R-D optimization and rate control together in a H.264 encoder. In Section 4.3, we introduce the different models that we used in the design of our proposed rate control scheme. The proposed rate control scheme is discussed in more details in Section 4.4. We present the experimental result in Section 4.5 before concluding in Section 4.6.

## 4.2 Rate-Distortion Optimization and Rate Control for H.264

Rate-Distortion optimization during video compression answers the question: "For each part of a video signal how and with what parameter settings should it be coded?" Though additional coding tools can improve coding performance of a codec, they complicate the decision making process during encoding.

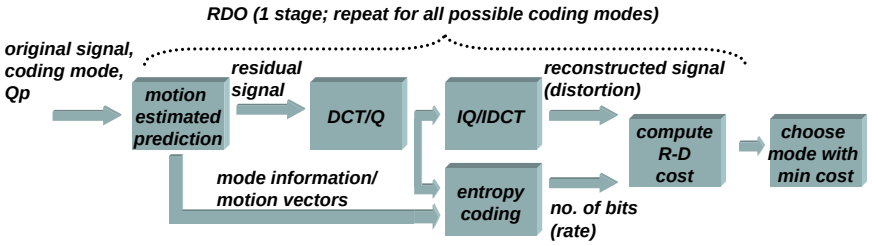



FIGURE 4.1: R-D optimization in a typical encoder

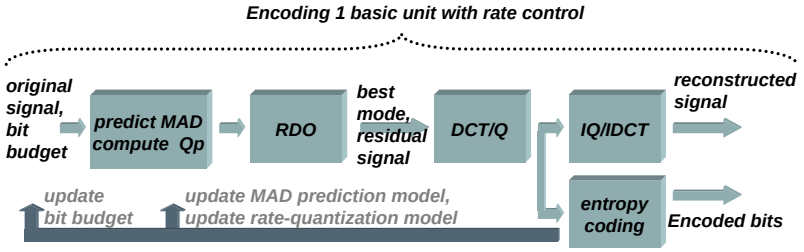



FIGURE 4.2: Encoding 1 basic unit with rate control in JM

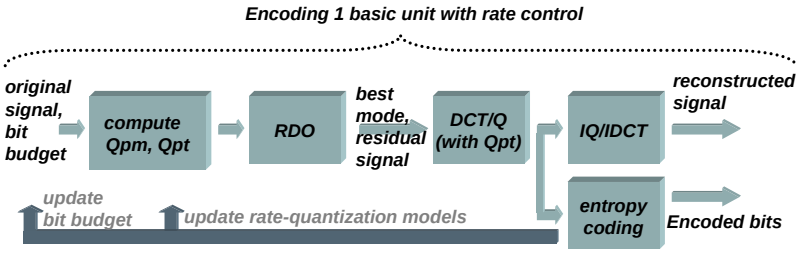



FIGURE 4.3: Encoding 1 basic unit with rate control with proposed scheme

Here, we briefly discuss the interaction between R-D optimization and rate control to highlight the difficulty of effectively implementing the two together in a H.264 encoder. Suppose that $Qp$ is the quantization parameter. During motion estimation, the encoder will select the motion vector, $\tilde{MV} = (mv_x, mv_y)$ that minimizes the following cost function:

$$J(\tilde{MV}, \lambda_{motion}) = SAD + \lambda_{motion} R(\tilde{MV}) \tag{4.1}$$

where SAD is the sum of absolute difference between original signal and the predicted signal, $R(\tilde{MV})$ is the number of bits required to code the motion vectors, and

$$\lambda_{motion} = 0.92 * 2^{(Qp-12)/6}. \tag{4.2}$$

The resulting distortion is weighted against the cost of coding the motion vectors to rule out those that are expensive to code but only bring marginal coding performance improvement.

After motion estimation is carried out for all available modes, the mode that leads to the lowest R-D cost is selected. This is usually done through minimizing the following cost

function:

$$J(mode, \lambda_{mode}) = SSD + \lambda_{mode} R(mode), \tag{4.3}$$

where *mode* is the MB mode that is chosen from the aforementioned set of defined MB modes, SSD is the Sum of Squared Difference, $R(mode)$ is the number of bits needed to code the MB and

$$\lambda_{mode} = \lambda^2_{motion} = 0.85 * 2^{(Qp-12)/3}. \tag{4.4}$$

If $J_k(mode_k, \lambda_{mode_k})$ is the Lagrangian cost function of the $k^{th}$ MB that is coded with mode $mode_k$ and **mode** is the $N$-tuple $(mode_0, ..., mode_{N-1})$, where $N$ is the total number of MB, a typical encoder assumes additive distortion and rate measures, resulting in the optimization problem:

$$\min_{\mathbf{mode}} \sum_{k=0}^{N-1} J_k(\mathbf{mode}, \lambda_{mode_k}) = \sum_{k=0}^{N-1} \min_{mode_k} J_k(mode_k, \lambda_{mode_k}). \tag{4.5}$$

That is, each MB selects the MB mode that gives the best R-D performance.

Fig. 4.1 shows the steps taken by an encoder to find out the distortion and the number of bits required when a particular mode is chosen for a MB in order to compute the R-D cost of the mode.

Since different coding modes perform differently at different bit-rates, a successful R-D optimized mode selection scheme has to consider the rate at which the video is encoded at; this is achieved through using $Qp$-dependent $\lambda$s in the Lagrangian formulations. Though this will improve coding performance over a range of bit-rates, it complicates the rate control process.

MAD-dependent rate-quantization models have been successful at capturing the relationship between $Qp$ and rate in DCT-based video coding system. However, the MAD is only available after R-D optimization has been carried out and the residual signal is available after motion-compensated prediction. To overcome this problem, the JM rate control scheme[78] assumes that some relationships exist between the MAD of co-located basic unit and attempts to predict the MAD of the encoded basic unit at the start of the encoding process (Fig. 4.2). The linear model used for the prediction is:

$$\mathtt{MAD}_{predicted} = Y_1 \mathtt{MAD}_{prev} + Y_2, \tag{4.6}$$

where $\mathtt{MAD}_{prev}$ is the MAD of the co-located basic unit of the previously encoded frame and $Y_1$ and $Y_2$ are model parameters. After predicting the MAD, a model such as:

$$R = X_1 \mathtt{MAD}/Qp + X_2 \mathtt{MAD}/Qp^2, \tag{4.7}$$

where $X_1$ and $X_2$ are model parameters that can be used to compute the $Qp$ before the R-D optimization. After the encoding of a basic unit, both the MAD-predicting model and the rate-quantization model are updated through linear regression with procedures to identify and exclude outliers.

It is possible that observing the residual signal after R-D optimization and then computing a $Qp$ value (can be different from the $Qp$ used during the R-D optimization) before the transform and quantization of the residual signal can improve rate control performance. This has been demonstrated in [79].

## 4.3 Modeling Rate and Distortion in H.264

In this work, we propose a simple model-based rate control scheme for operating a H.264 video encoder that uses R-D optimization for good coding performance. At the start of the encoding process of each basic unit, the target rate of the unit has to be determined. The $Qp$ with which the basic unit is coded can then be computed to meet the bit budget, $B$. To achieve this, we use models of texture bits, non-texture bits and distortion expressed in terms of $Qp_m$ and $Qp_t$, where $Qp_m$ is the $\lambda$-determining $Qp$ during R-D optimization and $Qp_t$ is used to determine the step-size of the quantizer. Using Lagrangian optimization, we derived values for $Qp_m$ and $Qp_t$ that minimize distortion given the rate constraint.

### 4.3.1 Rate-Quantization Model for Non-texture Bits

The proportion of non-texture bits in H.264 is significantly higher compared to previous standards. The increase can be largely attributed to the higher number of motion vectors. The option of using finer partitions can potentially improve prediction performance leading to a decrease in the amount of texture information that needs to be coded. This is at the expense of a higher number of non-texture bits. The superior coding performance of H.264 justifies the inclusion of the extra coding options.

During motion estimated prediction, an encoder can conceivably choose to use the finest partition mode available in the codec to provide the best prediction performance. Though the finest partition will bring about the best prediction, minimizing the power of the residual signal, the number of non-texture bits that have to be coded increases. If using a finer partition mode and suffering the penalty of a higher number of non-texture bits only brings about a marginal prediction performance, the overall coding performance may not improve.

Similarly, the determination of motion vectors cannot be carried out with the sole aim of minimizing the power of residual signal. If the total entropy of the resulting motion vectors is large, they may be too costly to code for the improvement in prediction performance they provide. To rule out such motion vectors, some mechanism must exist to weigh the prediction performance of motion vectors and the cost incurred when coding them (e.g. Equation 4.1).

Since the $Qp$ used during quantization affects mode and motion vector selection during R-D optimization, it is not surprising that the non-texture bits (used for coding mode decisions

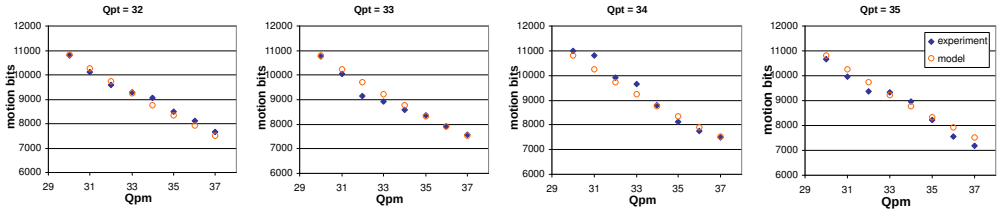



FIGURE 4.4: Comparing $R_m - Qp$ model and empirical data (FOOTBALL). Model: $R_m = \frac{c_1}{Qp_m} + c_2$.

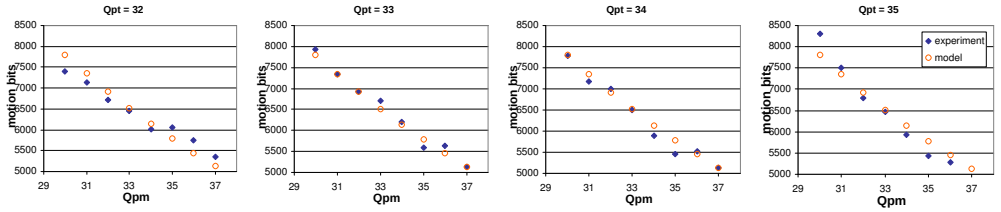



FIGURE 4.5: Comparing $R_m - Qp$ model and empirical data (SOCCER). Model: $R_m = \frac{c_1}{Qp_m} + c_2$.

and motion vectors) is dependent on $Qp$. We observed that generally the number of non-texture bits increases as $Qp_m$ decreases. We model the relationship between $Qp_m$ and the number of non-texture bits as follows:

$$R_m = \frac{c_1}{Qp_m} + c_2, \tag{4.8}$$

where $R_m$ is the number of non-texture bits and $c_1$ and $c_2$ are model parameters.

To verify the accuracy of the proposed model (4.8), we ran the encoder numerous times, varying $Qp_t$ and $Qp_m$ and observed the number of texture and non-texture bits of the encoded file. After the experiment, we fit the data to our models and computed the model parameters through linear regression. The experimental data are then plotted with the updated models to show the models' ability to capture the different relationships.

Figs. 4.4 and 4.5 show the experimental results and the models relating $R_m$ and $Qp_m$ obtained with the first P-frames of the sequences FOOTBALL and SOCCER.

### 4.3.2 Rate-Quantization Model for Texture Bits

After motion-compensated prediction, the residual signal in each basic block is transformed, quantized before being coded. Previous standards such as MPEG-1, MPEG-2, MPEG-4 and H.263 use the 8x8 Discrete Cosine Transform (DCT) as the basic transform. The main transform specified in H.264 that operates on 4x4 blocks of residual data is a multiplication-free integer transform that is based on the DCT.

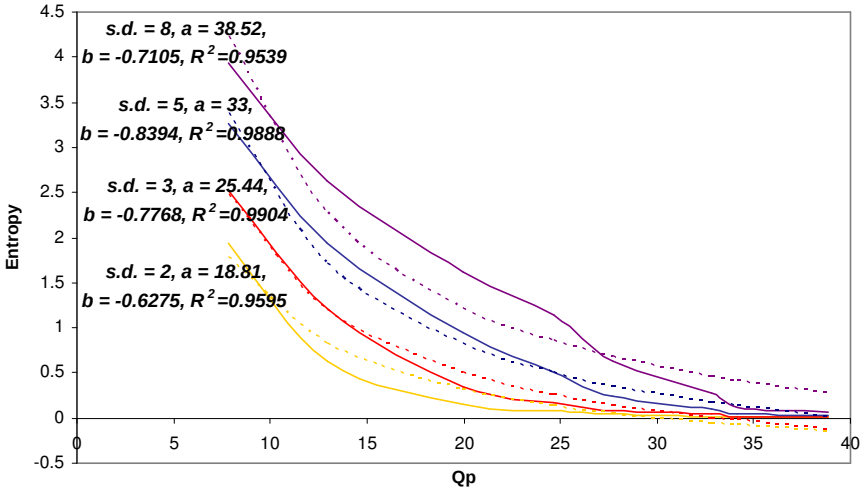



FIGURE 4.6: Entropy of Laplacian Distribution with $Qp$ when $\sigma = 2, 3, 5, 8$. Solid lines are obtained from (4.9). Dashed lines are obtained when $H(Qp_t) - Qp_t$ relationships are fitted to model, $H(Qp_t) = \frac{a}{Qp_t} + b$.

Lam and Goodman [80] showed that the Laplacian distribution shows good fidelity to empirical data when used to model DCT coefficients for images. Several previous works also modelled DCT coefficients of motion-compensated prediction residues with Laplacian distribution with variance $\sigma^2$[72][81]. In addition, [72] also approximated the entropy of a $Q$-quantized Laplacian, $H(Q)$ as:

$$H(Q) = \begin{cases} \frac{1}{2} log_2(2e^2 \frac{\sigma^2}{Q^2}), & \frac{\sigma^2}{Q^2} > \frac{1}{2e} \\ \frac{e}{ln2} \frac{\sigma^2}{Q^2}, & \frac{\sigma^2}{Q^2} \leq \frac{1}{2e} \end{cases} \tag{4.9}$$

In H.264, a total of 52 values of $Q$ are supported; these are indexed by the $Qp_t$. Through adjusting $Qp_t$, an encoder can directly control the trade-off between decoded video quality and bit-rate of encoded video. The relationship between $Q$ and $Qp_t$ is

$$Q = 2^{Qp_t/6} \nu(Qp_t mod 6), \tag{4.10}$$

where $\nu(0) = 0.675$; $\nu(1) = 0.6875$; $\nu(2) = 0.8125$; $\nu(3) = 0.875$; $\nu(4) = 1.0$; $\nu(5) = 1.125$.

Fig. 4.6 shows that the relationship between the entropy of quantized Laplacian random variable and $Qp_t$ can be reasonably modelled by a function of the form $H(Qp_t) = \frac{a}{Qp_t} + b$, where $a$ and $b$ are model parameters.

The value of $Qp_m$ also has effect on the eventual number of texture bits. Finer partitions are generally favoured at higher bit-rates to give better prediction. With better prediction, the power of the residual signal is reduced resulting in less texture bits. Therefore, we can expect the number of texture bits to increase when $Qp_m$ increases. However, at lower bit-rates, it is also more likely that more decisions to skip macroblocks will be made, decreasing the number

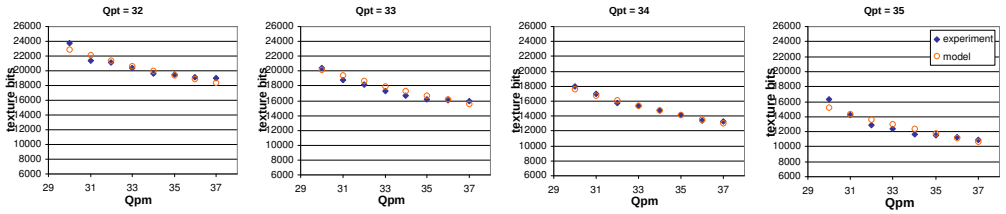



FIGURE 4.7: Comparing $R_t - Qp$ model and empirical data (FOOTBALL). Model: $R_t = \frac{b_1}{Qp_m} + \frac{b_2}{Qp_t} + b_3$.

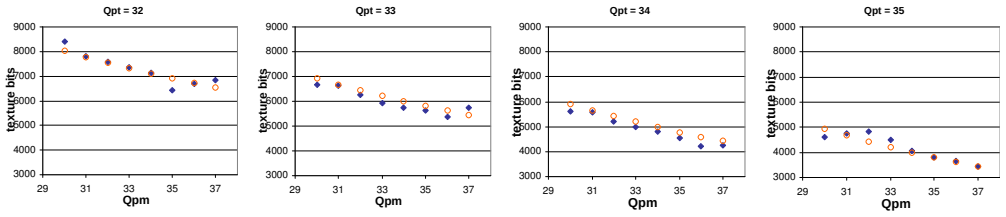



FIGURE 4.8: Comparing $R_t - Qp$ model and empirical data (SOCCER). Model: $R_t = \frac{b_1}{Qp_m} + \frac{b_2}{Qp_t} + b_3$.

of texture bits. Though it is more difficult to predict the relationship of the number of texture bits and $Qp_m$, our experiment on several sequences indicates an inverse relationship.

Considering the effect of both $Qp_t$ and $Qp_m$ on the number of texture bits, we model the relationship between texture bits and the two $Qp$s as:

$$R_t = \frac{b_1}{Qp_m} + \frac{b_2}{Qp_t} + b_3, \tag{4.11}$$

where $b_1$, $b_2$ and $b_3$ are model parameters.

As we have done in the previous subsection, we updated the model with empirical data collected from experiment and plotted the updated model and the experimental data together. It can be observed that our model provides a reasonable approximation of the texture bits-$Qp_t$-$Qp_m$ relationship. Figs. 4.7 and 4.8 show the experimental results and the updated models obtained with the first P-frames of the sequences FOOTBALL and SOCCER.

### 4.3.3 Distortion Model

Quantization of transform coefficents leads to distortion and affects the fidelity of the reconstructed video signal. In this work, we use a new model to approximate the relationship between distortion, $Qp_m$ and $Qp_t$. As a measure of distortion, we use the mean-absolute-difference, $D_{mad}$ between the original and reconstructed video signals. Through experiment, we found that $D_{mad}$ varies linearly with both $Qp_m$ and $Qp_t$ and thus can be modelled by:

$$D_{mad} = a_1 Qp_m + a_2 Qp_t + a_3, \tag{4.12}$$

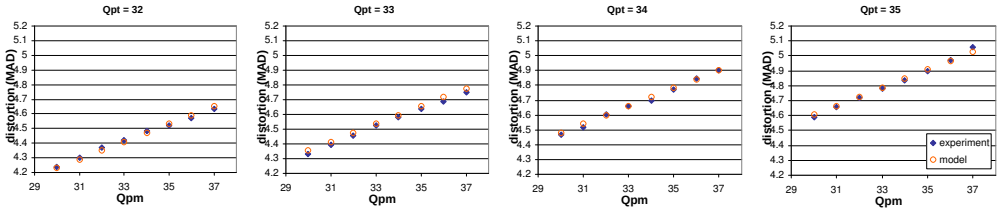



FIGURE 4.9: Comparing $D - Qp$ model and empirical data (FOOTBALL). Model: $D = a_1 Qp_m + a_2 Qp_t + a_3$.

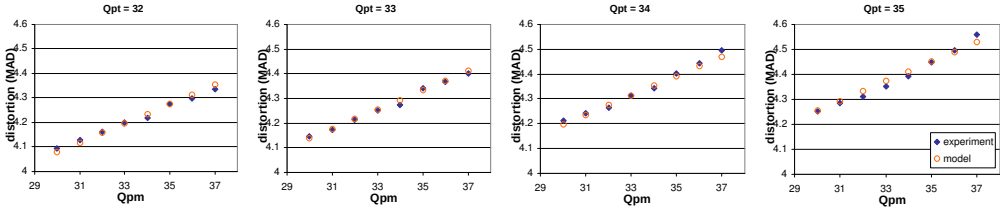



FIGURE 4.10: Comparing $D - Qp$ model and empirical data (SOCCER). Model: $D = a_1 Qp_m + a_2 Qp_t + a_3$.

where $a_1$, $a_2$ and $a_3$ are model parameters. Figs. 4.9 and 4.10 show the experimental results and the updated models obtained with the first P-frames of the sequences FOOTBALL and SOCCER.

There are other models proposed to approximate the rate-distortion-quantization behaviour of DCT-based encoder. For example, several work derived rate-quantization approximations based on classical rate-distortion models leading to logarithmic expressions [82][83]. When choosing models for the design of a rate control scheme, there has to be a balance among fidelity to empirical data, tractability of the eventual optimization problem and the ease of model updates. In the next section, we will show how the proposed models enable us to design a simple yet effective rate control scheme for H.264.

## 4.4 Rate Control

In this section, we use the model described in the previous section to design a rate control scheme that is effective for a H.264 encoder that also implements R-D optimization (Fig. 4.2). The main steps of the rate control scheme are: bit allocation, $Qp$s determination and model update.

### 4.4.1 Bit Allocation

We use the same bit allocation method used in JM rate control to determine the number of bits assigned to a basic unit at the beginning of its encoding. When allocating bits from a

fixed bit budget to basic units in a video frame, the relative complexity of each basic unit has to be considered for good coding performance.

Consider the computation of target number of bits, $B_{l,i}$ for the $l^{th}$ basic unit of the $i^{th}$ frame. Let $\tilde{\texttt{MAD}}_{l,i}$ be its predicted mean absolute difference. The prediction is carried out using the actual MADs of co-located basic unit in previously coded frames. Specifically, the number of target bits for the $l^{th}$ basic unit is given by:

$$B_{l,i} = T_{r,i} \frac{\tilde{\texttt{MAD}}_{l,i}^2}{\sum_{k=l}^{N} \tilde{\texttt{MAD}}_{k,i}^2}, \tag{4.13}$$

where $T_{r,i}$ is the remaining number of bits in the frame $i$'s bit budget left for coding the $l^{th}$ and other uncoded basic unit in frame $i$. $N$ is the total number of basic units in frame $i$.

### 4.4.2 Quantization Parameters Selection with Lagrangian Optimization

The optimization process in our rate control scheme follows a simple framework. With the models $D = f_D(Qp_t, Qp_m)$, $R_t = f_t(Qp_t, Qp_m)$ and $R_m = f_m(Qp_m)$. The Lagrangian optimization formulation for the $l^{th}$ basic unit of the $i^{th}$ frame is in the form: $min(D)$ given that $R_t + R_m \leq B_{l,i}$.

With the models defined in (4.8),(4.11) and (4.12), the optimization problem is:

minimize

$$a_1 Qp_m + a_2 Qp_t + a_3 \tag{4.14}$$

given that

$$\frac{b_1}{Qp_m} + \frac{b_2}{Qp_t} + b_3 + \frac{c_1}{Qp_m} + c_2 < B_{l,i}. \tag{4.15}$$

Solving the constrained optimization problem yields the following expressions for $Qp_t$ and $Qp_m$:

$$Qp_m^* = Qp_t^* \sqrt{\frac{a_2(b_1 + c_1)}{a_1 b_2}} \tag{4.16}$$

and

$$Qp_t^* = \frac{\sqrt{\frac{a_1 b_2 (b_1 + c_1)}{a_2}} + b_2}{B_{l,i} - c_2 - b_3}. \tag{4.17}$$

### 4.4.3 Model Update

Since the characteristics of an encoded video sequence can vary as encoding progresses, a rate control scheme can be more effective if it can adapt the models it uses throughout the encoding process. For example, the rate control scheme implemented in JM reference software updates the models it uses to approximate the rate-quantization relationship and to predict the MAD of a basic unit before encoding.

In our rate control scheme, we only update two models parameters $c_2$ and $b_3$. The rest of the parameters are empirically determined and maintained at the same values for all sequences in the experiment. After encoding of the $l^{th}$ basic unit of the $i^{th}$ frame, $c_2(l+1,j)$ and $b_3(l+1,j)$ the updated model parameters for the next basic unit is:

$$c_2(l+1,j) = R_m(l,j) - \frac{c_1}{Qp_m(l,j)} \tag{4.18}$$

and

$$b_3(l+1,j) = R_t(l,j) - \frac{b_1}{Qp_m(l,j)} + \frac{b_2}{Qp_t(l,j)}, \tag{4.19}$$

where $Qp_t(l,j)$ and $Qp_m(l,j)$ are the $Qp$s used by the $l^{th}$ basic unit and $R_m(l,j)$ and $R_t(l,j)$ are the resulting motion and texture rates.

It is conceivable that more sophisticated model update procedures may improve rate control performance. For example, the encoder can carry out a linear regression with a set of $Qp$s and resulting rates collected from previously coded basic units. It is also possible for the encoding to encode selected basic units repeatedly, varying $Qp$ values to collect enough data for model parameters update.

The proposed rate control scheme is effectively computing a new lagrange multiplier after the encoding of each coding unit. Though this only guarantees optimal rate allocation among MBs in the same basic unit, this helps the encoder meet the bit target more effectively. Experiment results will some that the accurate rate control does not come at an expense of poorer reconstructed quality.

We resolve the RDO/rate control conundrum with a method that is different from the scheme proposed by Kwon et al [79]. The work by Kwon et al achieves rate control in the following steps:

- Stage 1: Carry out RDO on all MBs in the frame using the average $Qp$ used in the previous frame.
- Estimate the header bit-rate using result of the previous step.
- Compute remaining bits available for transform coefficients.

- Stage 2: Code each MB using rate-quantization model.
- Update model parameters with statistics from past 5 frames.

Kwon et al's 2-stage method requires one additional forward and inverse transform and one more quantization process per 4x4 block. Prediction of INTRA MBs also has to be recomputed. The proposed scheme achieves rate control in the following steps:

- Compute $Qp_m$ and $Qp_t$ simultaneously using (4.16), (4.17).
- Update $b_3$ and $c_2$ using statistics from most recently encoded basic unit.

Other than achieving effective rate control, there are two other possible benefits of the proposed rate control framework:

- It is likely that development of future codec will move in the direction of providing more prediction modes to achieve better prediction performance. When prediction performance improves, the power of prediction residual signal will decrease while the number of bits required for header information and motion vectors will increase.

  As the proportion of texture bits decreases, the approach of modelling texture and non-texture bit separately to determine $Qp_t$ and $Qp_m$ can be more effective than merely determining the quantization parameter by observing the prediction residual after R-D optimization.

- When encoding layered video, the encoder has to optimize the video quality over a range of bit-rates. A typical way of achieving signal-to-noise ratio scalability is to refine the values of transformed coefficients with layers of data that increase the precision of coefficients [66]. More layers of data can be transmitted and decoded when better video quality is desired.

  Since mode decisions and motion estimation can only be optimized at a certain bitrate, the motion vector field that is used will not be optimal at all extractable bit-rates of a layered video. When the motion vector field acquired is optimized at the base layer by carrying out R-D optimization with $\lambda$ determined with the base layer quantization parameter, $Qp_{base}$, the motion vector field will not be optimized for the enhancement layer that is coded with a smaller $Qp$. The coding performance at the enhancement layer will be compromised as a result.

  Though the motion vector field can only be optimized at one bit-rate, a possible way to achieve trade-off between 2 layers of video is to optimize the base layer motion vector field with a lower $Qp$ than $Qp_{base}$:

$$\lambda_{mode} = \lambda_{motion}^2 = 0.85 * 2^{(Qp_{base}-\beta-12)/3}, \tag{4.20}$$

where $\beta$ is a positive constant that can be used to control the coding performance trade-off between two successive layers.

In the proposed rate control framework, since $Qp_m$ and $Qp_t$ can be different, a scheme that takes into consideration the coding performance trade-off between two successive layers can be designed.

## 4.5 Experimental Results

We implemented and compare our rate control scheme with the scheme implemented JM 13.2 by experimenting on several sequences. We use the same method to determine the $Qp$s of the first INTRA frame and first P frame. The model parameters used during encoding are shown in Table 4.1. As described in sub-Section 4.4.3, only parameters $b_3$ and $c_2$ are updated according to (4.19) and (4.18).

We present the bits required to code each frame and also the average PSNR of the entire sequence (Fig. 4.11 - Fig. 4.22). Two sets of experiments were carried out with each sequence. First, both the JM 13.2 rate control and the proposed rate control scheme were used to control the bit-rate of the encoded video at a particular value. Another set of experiment is carried out where the target bit-rate changes in the middle of the encoding process. The second set of experiments shows the rate control scheme's ability to encode at different bit-rates and also its capability of adapting to changing channel conditions.

The standard deviation of the number of frame encoding bits for each sequence is shown on the plot of its constant bit-rate experiment (also summarized in Table 4.2). The set of tested sequences are chosen to include contents with both high and variable motion (CREW, FOOTBALL, SOCCER) and contents with comparatively lower motion activities (SILENT, AKIYO, MOTHER AND DAUGHTER). From the experimental results, we can observe that the proposed rate control scheme is able to achieve better rate control (reducing variability from target bit-rates) for all tested sequence. For more complex sequences like SOCCER, FOOTBALL and CREW, the proposed rate control scheme displayed better rate control while maintaining PSNR performance (within 0.15dB). For less complex sequences like AKIYO and MOTHER AND DAUGHTER, the proposed scheme showed PSNR gain of 0.5-0.7dB over JM rate control on top of better rate control performance.

Though the rate-quantization and distortion-quantization relationship can be approximated by the proposed models, the model parameters for different sequences (and different frames of the same sequence) can be very different (Figs. 4.4-4.10), so it is rather surprising that using mostly the same parameters and updating $b_3$ and $c_2$ resulted in satisfactory rate control performance in all tested sequence. Updating $b_3$ and $c_2$ with data collected from the previously encoded basic unit provides a fast approximation of the characteristics of the current coded basic unit. This simple update method's ability to track source characteristics overcomes

| model parameter | $a_1$ | $a_2$ | $a_3$ | $b_1$ | $b_2$ | $b_3$ | $c_1$ | $c_2$ |
|---|---|---|---|---|---|---|---|---|
| | 0.0604 | 0.1252 | -1.5953 | 72040 | 268090 | updated (4.19) | 52164 | updated (4.18) |

TABLE 4.1: model parameters: 36 basic units per CIF frame (for different number of basic units and/or resolution $b_1$, $b_2$, $c_1$ and $c_2$ are scaled accordingly). **Same set of parameters and update procedures are applied to all tested sequence.**

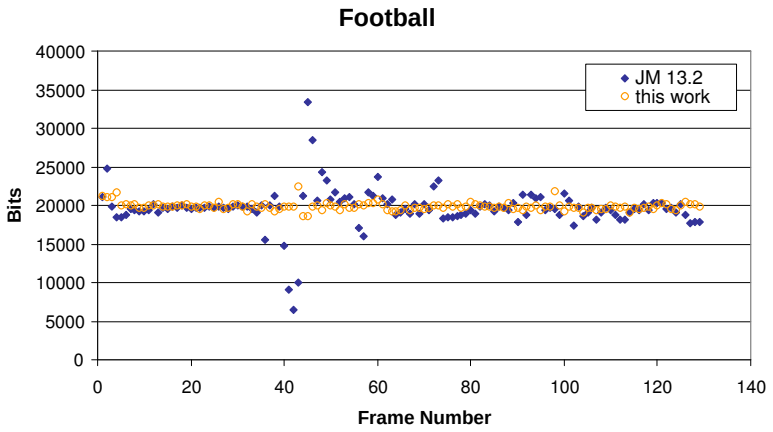



FIGURE 4.11: Football Rate Control CIF, 15fps (Target bitrate 300$kbps$, 36 basic units per frame). Average PSNR - JM13.2: 31.31dB, this work: 31.35dB. Encoding bits standard deviation - JM13.2: 2651, this work: 502.

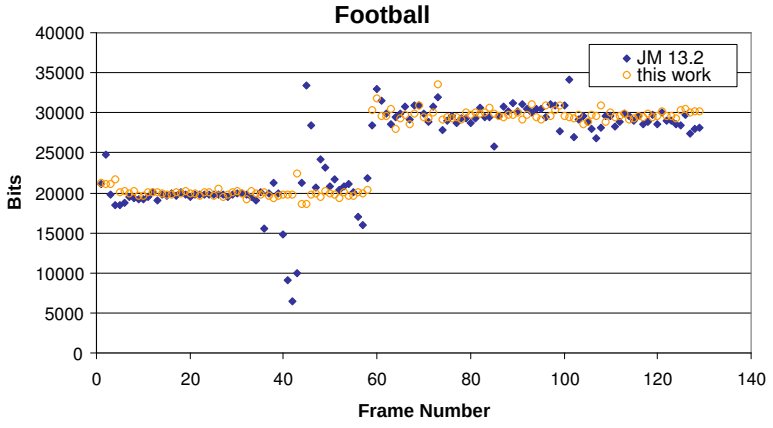



FIGURE 4.12: Football Rate Control CIF, 15fps (Target bitrate 300$kbps$, 450$kbps$, 36 basic units per frame). Average PSNR - JM13.2: 32.42dB, this work: 32.34dB.

model inaccuracies resulting from updating only two parameters ($b_3$ and $c_2$), resulting in a rate control scheme that outperforms the JM rate control scheme. Regression-based model update techniques that are used in most rate control schemes are more complex and often suffer from slow model adaptation as the success of adaptation is dependent on the statistics of several previously coded units. Hence, the performance of these schemes often suffer when used on sequences with fast varying source characteristics.

The rate control scheme's ability to track the source data characteristics can be most clearly observed when encoding the sequence FOOTBALL (Figs. 4.11-4.12). A rapid pan when the camera attempts to track the football as it is being passed from one player to another causes

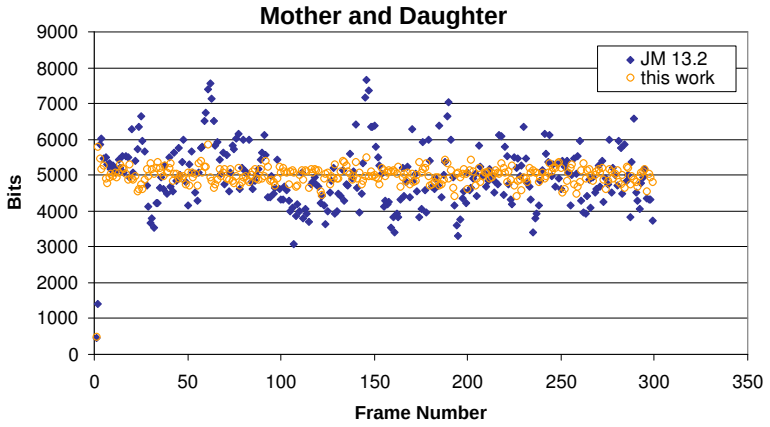



FIGURE 4.13: Mother and Daughter Rate Control CIF, 15fps (Target bitrate 75*kbps*, 66 basic units per frame). Average PSNR - JM13.2: 39.5dB, this work: 39.97dB. Encoding bits standard deviation - JM13.2: 831 , this work: 216.

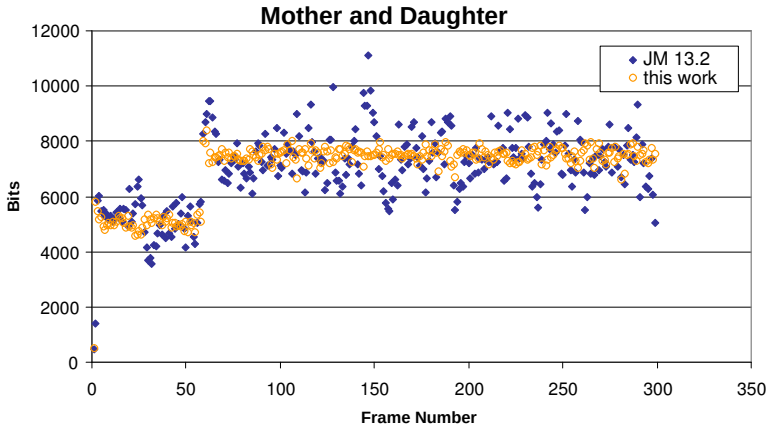



FIGURE 4.14: Mother and Daughter Rate Control CIF, 15fps (Target bitrate 75*kbps*, 112.55*kbps*66 basic units per frame). Average PSNR - JM13.2: 40.69dB, this work: 41.11dB.

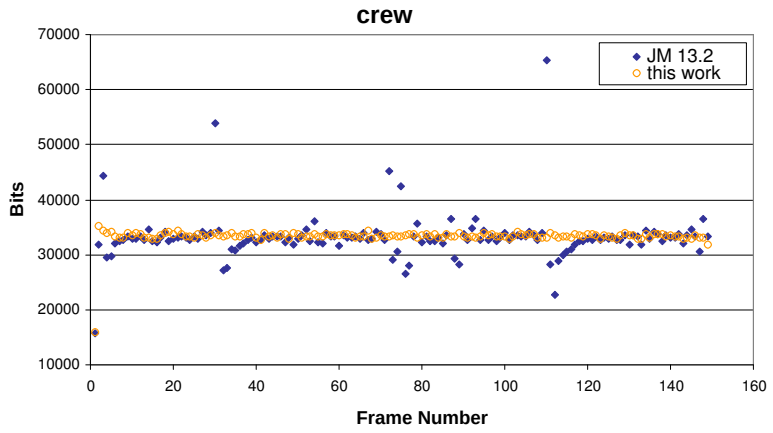



FIGURE 4.15: Crew Rate Control CIF, 15fps (Target bitrate 500*kbps*, 36 basic units per frame). Average PSNR - JM13.2: 37.52dB, this work: 37.51dB. Encoding bits standard deviation - JM13.2: 3983 , this work: 397.

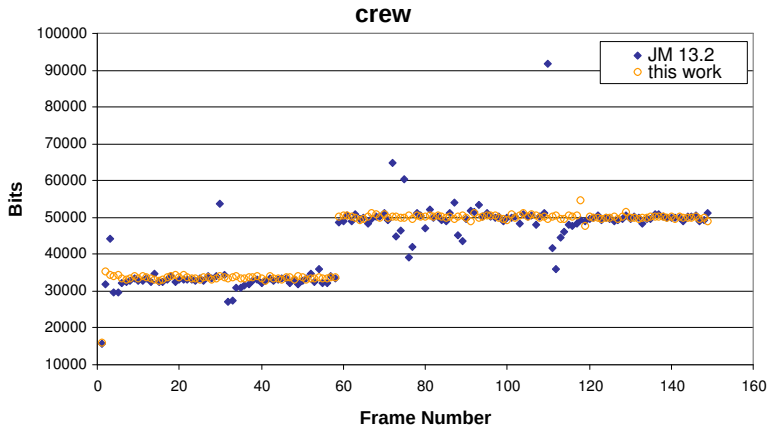



FIGURE 4.16: Crew Rate Control CIF, 15fps (Target bitrate 500*kbps*, 750*kbps*, 36 basic units per frame). Average PSNR - JM13.2: 38.73dB, this work: 38.67dB.

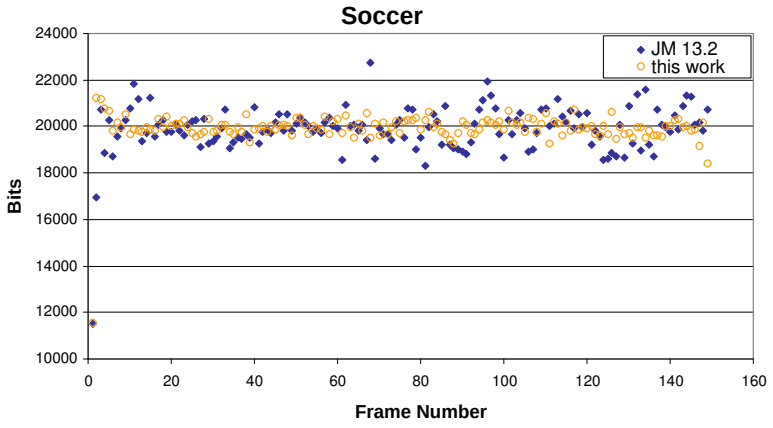



FIGURE 4.17: Soccer Rate Control CIF, 15fps (Target bitrate 300*kbps*, 66 basic units per frame). Average PSNR - JM13.2: 34.78dB, this work: 34.71dB. Encoding bits standard deviation - JM13.2: 802 , this work: 368.

the JM encoder's bit-rate to fluctuate. In contrast, the proposed rate control scheme is able to adapt quickly to the changing source video characteristics and produce bit-stream of near constant bit-rate.

The experimental results demonstrate the effectiveness of the proposed rate control scheme show that *a set of well selected curves, translated along their vertical axis when necessary, can sufficiently approximate the underlying rate-distortion-quantization relationships of most sequences to provide fairly good rate control performance*.

Figs. 4.23-4.25 show the performance of the proposed scheme relative to the work in [84] that attempted to improve rate control performance by improving MAD prediction prior to R-D optimization. Other than showing the performance improvement of the proposed scheme over a more recent piece of work, the comparisons also give indication that improving MAD prediction performance may not significantly improve rate control performance of JM rate control scheme.

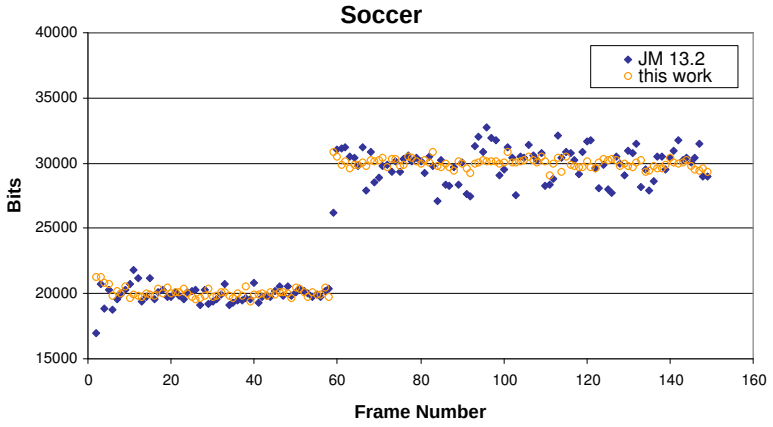



FIGURE 4.18: Soccer Rate Control CIF, 15fps (Target bitrate 300*kbps*, 450*kbps*, 66 basic units per frame). Average PSNR - JM13.2: 35.93dB, this work: 35.78dB.

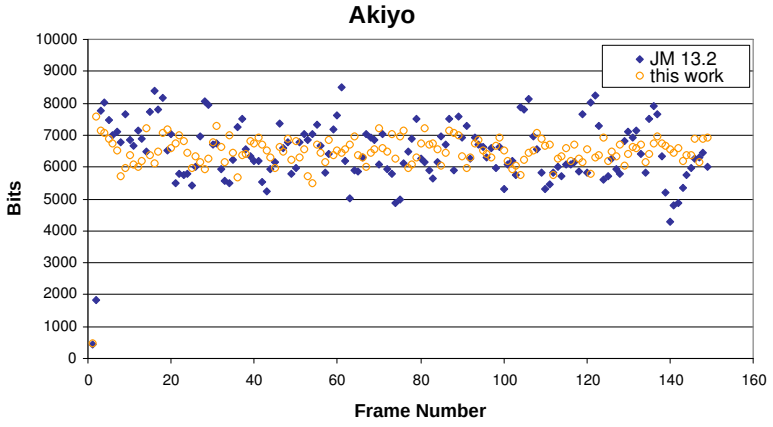



FIGURE 4.19: Soccer Rate Control 4CIF, 15fps (Target bitrate 100*kbps*, 264 basic units per frame). Average PSNR - JM13.2: 41.42dB, this work: 42.15dB. Encoding bits standard deviation - JM13.2: 925 , this work: 383.

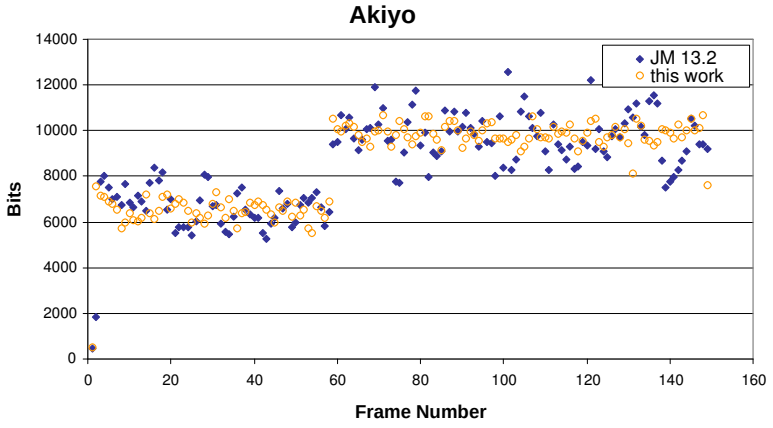



FIGURE 4.20: Soccer Rate Control 4CIF, 15fps (Target bitrate 100*kbps*, 150*kbps*, 264 basic units per frame). Average PSNR - JM13.2: 42.07dB, this work: 42.77dB.

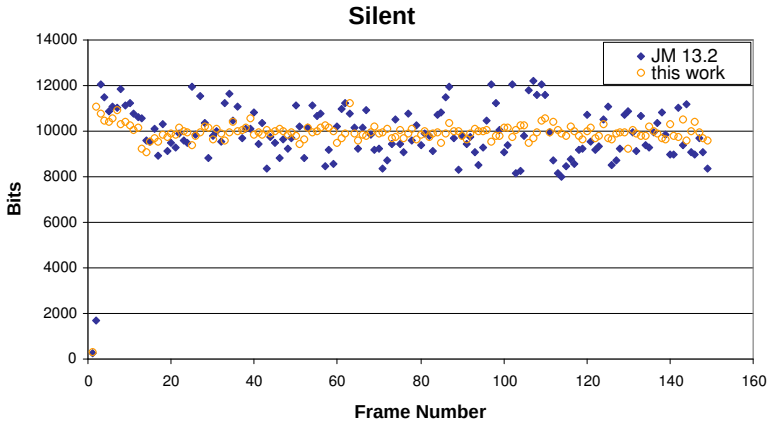



FIGURE 4.21: Silent: Rate Control CIF, 15fps (Target bitrate 150*kbps*, 66 basic units per frame). Average PSNR - JM13.2: 36.95dB, this work: 37dB. Encoding bits standard deviation - JM13.2: 1226 , this work: 323.

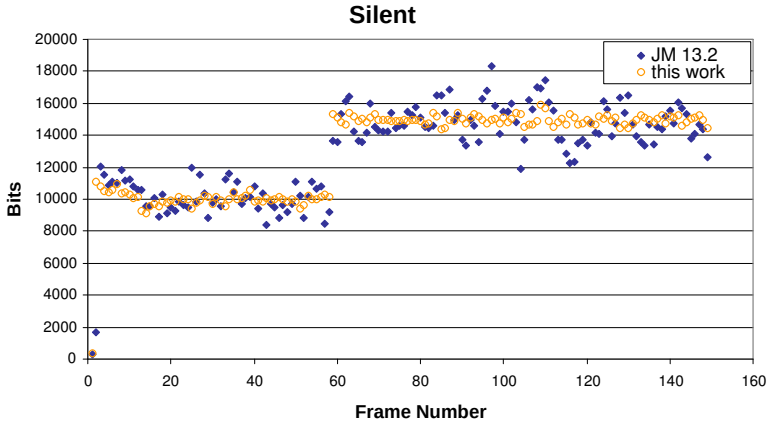



FIGURE 4.22: Silent: Rate Control CIF, 15fps (Target bitrate 150*kbps*, 225*kbps*, 66 basic units per frame). Average PSNR - JM13.2: 37.95dB, this work: 37.99dB.

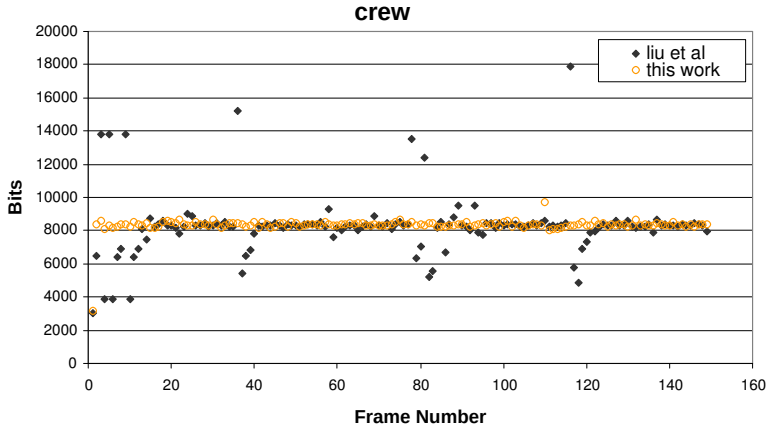



FIGURE 4.23: Crew: Rate Control CIF, 15fps (Target bitrate 125*kbps*, 66 basic units per frame). Average PSNR - Liu et al: 31.4dB, this work: 31.57dB. Encoding bits standard deviation - Liu et al: 1682 , this work: 164.

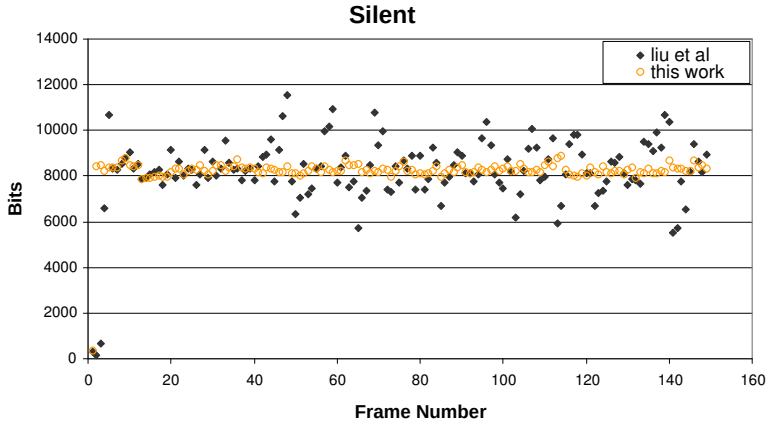



FIGURE 4.24: Silent: Rate Control CIF, 15fps (Target bitrate 125*kbps*, 66 basic units per frame). Average PSNR - Liu et al: 36.16dB, this work: 36.13dB. Encoding bits standard deviation - Liu et al: 1104 , this work: 164.

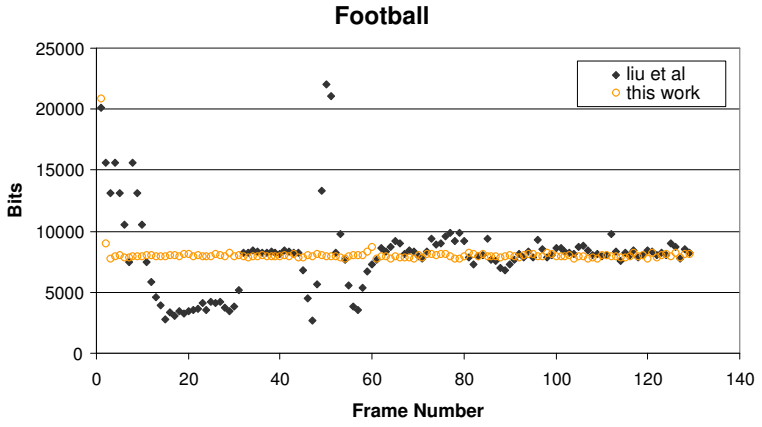



FIGURE 4.25: Football: Rate Control CIF, 15fps (Target bitrate 125*kbps*, 66 basic units per frame). Average PSNR - Liu et al: 27.37dB, this work: 27.57dB. Encoding bits standard deviation - Liu et al: 3012 , this work: 173.

Figs. 4.26 and 4.27 show that the proposed rate control scheme does not result in PSNR fluctuation that is significantly different from the reference rate control scheme.

## 4.6 Remarks

Implementing both R-D optimization and rate control in a H.264 video encoder can be rather challenging. In this chapter, we proposed a new model-based approach to achieve effective rate control for H.264-complaint encoders. By modelling the rate-distortion-quantization relationships for both texture and non-texture bits, we were able to compute values for $Qp$ used for R-D optimization and quantization of transformed coefficients through Lagrangian optimization. This, when coupled with a straightforward update procedure, enables us to build

TABLE 4.2: Results Summary (for constant bit-rate experiment)

| | average PSNR (dB) | | encoding bits stdev | |
|---|---|---|---|---|
| | JM | proposed | JM | proposed |
| CREW | 37.52 | 37.51 | 3983 | 397 |
| FOOTBALL | 31.31 | 31.35 | 2651 | 502 |
| SOCCER | 34.78 | 34.71 | 802 | 368 |
| MOTDAU | 39.5 | 39.97 | 831 | 216 |
| SILENT | 36.95 | 37 | 1226 | 323 |
| AKIYO | 41.42 | 42.15 | 925 | 383 |

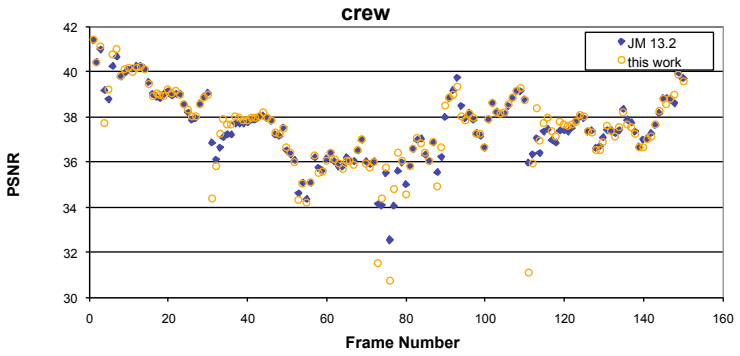



FIGURE 4.26: Crew: PSNR fluctuation.

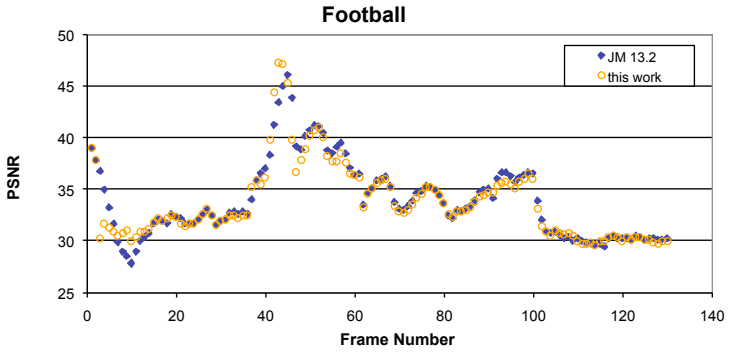



FIGURE 4.27: Football: PSNR fluctuation.

a simple yet effective rate control scheme that adapts quickly to changing source data characteristics. When compared to JM rate control, our proposed rate control scheme significantly decreases encoding bits variability with comparable or better video quality.

Following this chapter, we will show that, other than being a useful tool, rate control, together with complexity control, provides flexibility during the optimization of a video encoding and transmission system that will lead to better overall performance.

# Chapter 5

# Complexity-Rate-Distortion Analysis for Constrained Encoding and Transmission

## 5.1 Introduction

In this chapter, we analyze the complexity-rate-distortion characteristics of the video encoder we described in Chapter 3. Through modeling its complexity-rate-distortion relationships, we derive optimized operating mode of the encoder (rate and complexity) and show through experiment that such optimization can help a video encoder operate within rate and complexity constraints. The design of the complexity scalable encoding scheme enables the encoder to perform optimization while taking into consideration the availability of computational resource. This extension of traditional rate-distortion optimization is necessary when time or power constraints do not allow a video encoder to achieve rate-distortion optimized coding performance. We also consider the selection of complexity and rate controlling source coding parameters jointly with transmitter power. Power invested during source encoding can potentially improve coding efficiency while transmission power determines the capacity of the delivery channel. A complexity scalable source encoder is a useful tool during overall system optimization as it allows flexible allocation of resource between the encoder and transmitter.

Progress in video coding research has significantly improved the coding performance of modern video coding schemes. Unfortunately, the various techniques that lead to better coding efficiency also increase the complexity of the video encoder. For block-based coding scheme, the encoder has to evaluate a number of possible coding modes to determine the optimal representation for each MB. In order to fully exploit the features to achieve optimal rate-distortion (R-D) performance, the encoder has to check all possible coding modes for each MB before selecting the best set of coding parameters.

The computational complexity of modern video encoders makes the design of real-time system a challenge. Real-time encoding is required for applications such as live broadcast, surveillance or video communication. Considering that these applications may be built on a wide variety of computing platforms, without an effective complexity scalable solution, it is extremely difficult for an encoder to fully utilize available resource while fulfilling real-time requirement. Integrating both source signal compression and radio transmission for time-constrained video transmission is even more of a challenge. This is especially true for mobile devices that are operating under power constraint due to either limited computational power or restricted battery life. To evaluate the importance of a complexity scalable encoder during system optimization under power constraint, we consider the situation where the power allocation between source encoder and transmitter can be calibrated. Since the transmitter power determines the transmission channel capacity, we attempt to maximize system performance by choosing encoder parameters, transmitter power and the encoded video bitrate optimally.

Several studies have been done to study the possibilities of joint optimization of different components in a video delivery system. The work in [85] attempted to model the entire video delivery system to arrive at a power efficient solution for multimedia communication over wireless channel. The work in [86] and [87] jointly optimized source coding and transmission power under bandwidth constraint but did not consider the possibility of encoder power control.

In this chapter, we study the power allocation between the source encoder and transmitter (the most power consuming parts of a real-time video delivery system). A key difference between our work and those in [88][89][85] and [90] is the way coding performance-complexity trade-off is achieved. The work in [90] defines five fixed encoder operating modes and corresponding coded modulation configurations. Lu et al [85] controls the complexity of video encoding by controlling the INTRA rate while [88] and [89] controls the motion estimation process. This work uses a singularly-parameterized complexity scalable rate-distortion framework for video encoders. With a single parameter $\beta$, the complexity-coding performance of our encoder can be controlled over a large range, it is also relatively easy to model its behaviour as the parameter changes. When $\beta$ is set to zero, the complexity and coding performance is that of an encoder that only implements the SKIP and INTRA modes. When $\beta$ is set to two, the complexity and coding performance is that of an encoder that does exhaustive mode decisions. Modeling the power consumption or complexity of video encoders for the purpose of system optimization has been proposed in [85] and [91].

Since the complexity of the encoder can be controlled with a single parameter, it is also relatively easy to model its behaviour as the parameter changes. In our work, we model the complexity of our video encoder and show how the complexity scalable feature can be useful in the design of an encoder operating under both complexity and rate constraints. Our optimization scheme outputs parameters that allow the encoder to be as close to being rate-distortion optimized as possible.

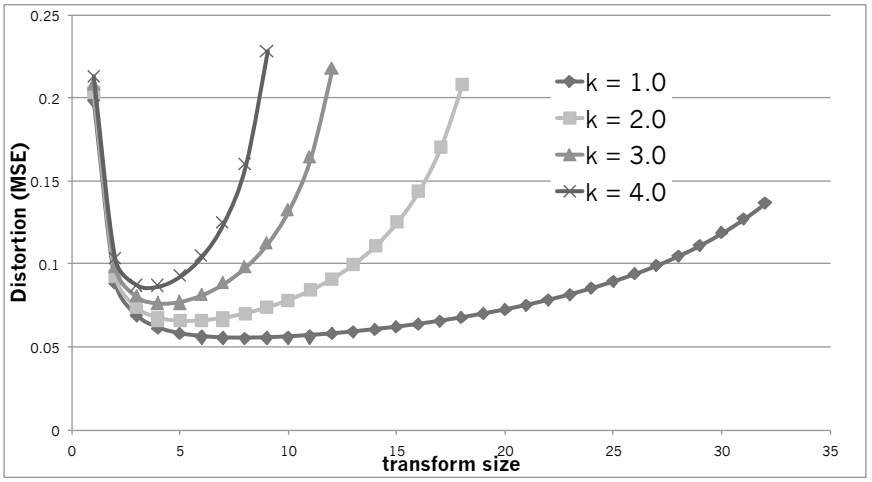



FIGURE 5.1: Operating under power constraint: Encoding and transmitting Gauss-Markov source (with varying $c$, $c = k \times 10^{-7}$)

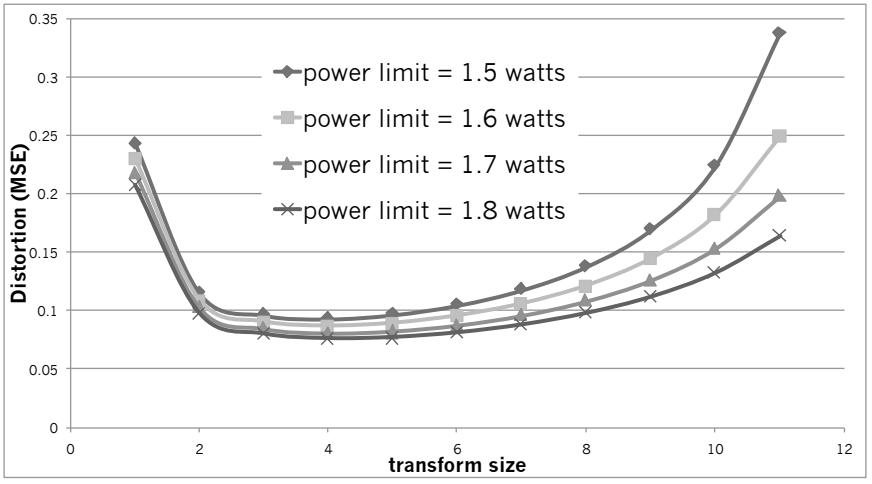



FIGURE 5.2: Operating under power constraint: Encoding and transmitting Gauss-Markov source (with varying $c$, $c = k \times 10^{-7}$)

## 5.2 Source encoder and transmitter power allocation

In this chapter, we are primarily concerned with the effect of adjusting the power allocation between the source encoder and transmitter. Since power invested in the source encoder can potentially improve coding performance and reduce the rate for a fixed reconstruction fidelity, while power channelled to the transmitter increases the capacity of the transmission channel, it is not obvious how power should be allocated between the source encoder and transmitter, or if adjusting the allocation will bring significant improvement.

We first consider the coding of first order Gauss-Markov source with variance $\sigma_s$ and auto-correlation function $\sigma_s^2|\rho|^k$, where $\rho$ is the correlation coefficient. Using a transform of size $n$,

the operational distortion-rate function using the Karhunen-Loeve Transform is

$$D(R_s) = \epsilon\sigma_s^2(1-\rho^2)^{(n-1)/n}2^{-2R_s}, \tag{5.1}$$

where $\epsilon$ is quantizer dependent and $R_s$ is the number of bits per sample. Using the transform size for complexity control was suggested in [85] and [68] . Let the capacity of the Gaussian channel be

$$C = \frac{1}{2}log(1+\frac{hp_t}{N_0W}), \tag{5.2}$$

where $P_t$ is the transmission power, $W$ is the bandwidth of the channel, $N_0$ is the noise power spectral density at the receiver and $h$ is attenuation. We assumed a channel code rate, $r$ of 0.5 and that the channel code is sufficient to ensure error-free transmission. The data rate available for the source encoder must then be reduced to $R_s = rC$. Since transforming a vector of dimension $N$ requires $N^2$ operations, assuming that the source is sampled at $f$ sample/s and that the power dissipated is proportional to the number of operations with the proportionality constant $c$, the power consumption of the source coder is $P_s = cfN$.

Using $N_0W = 0.39$, $W = 500kHz$ [92], $\sigma^2 = 1$, $h = 0.9$, $\rho = 0.9$ and $\epsilon = 1$, we present 2 sets of numerical examples to show the benefits of controlling source encoder and transmitter jointly (Figs. 5.1 and 5.2). Fig. 5.1 shows the effect of transform size selection on distortion for a fixed power limit. When the cost of each operation is high (higher $c$), the penalty on the system when over-investing on the encoder (large $N$ is chosen) is heavier. For an encoder of fixed operation cost, the problem of choosing an overly large transform size is more obvious at low power limit (Fig. 5.2).

From the above study of the transform coding of Gauss-Markov source, we observe that, depending on the relative power consumption of the source encoder and transmitter, there is significant incentive to design a complexity (power) scalable source encoder. In subsequent sections, we will study the power allocation between a power scalable video encoder and the wireless transmitter using an empirically-derived complexity-rate-distortion model.

## 5.3 Complexity-Rate-Distortion Analysis of Video Encoder

Although the analysis in the previous section provided some insights on the benefits of optimizing the source encoder and transmitter jointly when the system is operating under power constraint, a modern video encoder consists of much more than the transform coding. Here, we study the complexity-rate-distortion behaviour of the encoder we described in Chapter 3. We encode 1 sec of a typical video sequence (SOCCER) and observed the complexity-rate-distortion behaviour of the encoder. The sequence is encoded as 1 I-frame followed by 15 P-frames.

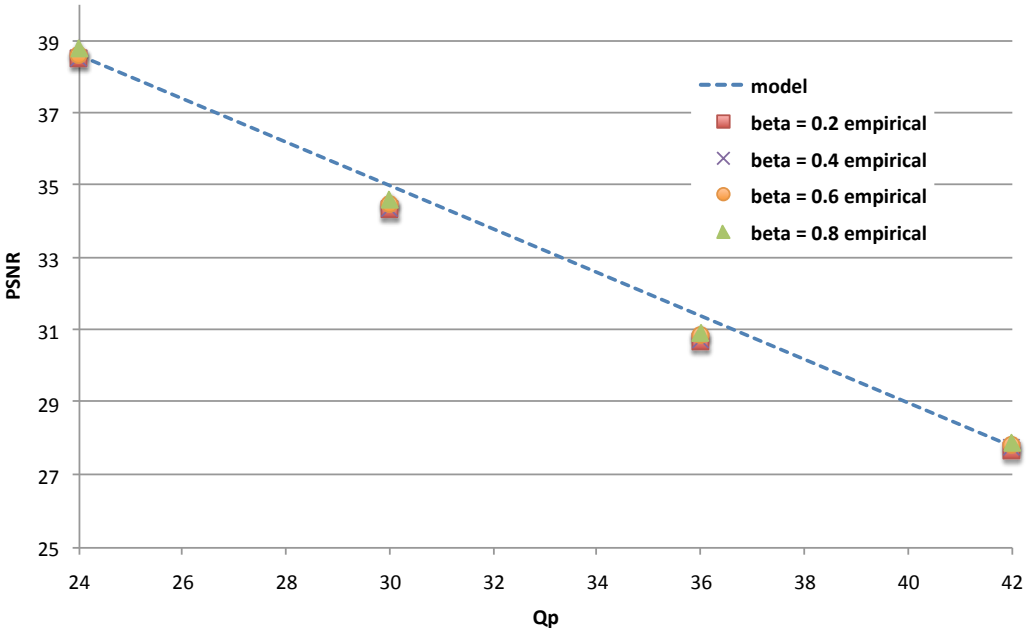



FIGURE 5.3: soccer: comparing actual PSNR and fitted model.

### 5.3.1 Distortion model

The linear relationship of $Q_p$ and PSNR is not surprising (Fig. 5.3). We observed that the distortion is not sensitive to $\beta$; coding performance improvement (in the low-delay settings) as $\beta$ is increased manifested itself as bit-rate reduction. Hence, we model the relationship of $\beta$, quantization parameter ($Q_p$) and distortion ($D_{PSNR}$) by

$$D_{PSNR}(Q_p, \beta) = -0.6Q_p + \gamma_D, \tag{5.3}$$

where $\gamma_D$ is a source dependent model parameter. We also found that most sequences display the same behaviour. This observation will be helpful when we try to design a rate and complexity control scheme for video encoding.

### 5.3.2 Rate Model

The rate and $Q_p$ follows a quadratic relationship. We observed that the number of bits required to represent the encoded video is insensitive to $\beta$ at low rate. We picked a high $Q_p$ ($Q'_p$=42, $r'$=90000) point and assume the $R$-$Q_p$ curves for all $\beta$ across all sequences pass through this point. Relationship between $R$ and $Q_p$ can be represented by an equation in the form of $R = A(Q_p - Q'_p)^2 + r'$. Observing that the relationship between coefficient $A$ and $\beta$ can be modeled by a quadratic function, we model the $R$-$Q_p$-$\beta$ relationship as

$$R(Q_p, \beta) = (\sigma_2\beta^2 + \sigma_1\beta + \sigma_0)(Q_p - Q'_p)^2 + r', \tag{5.4}$$

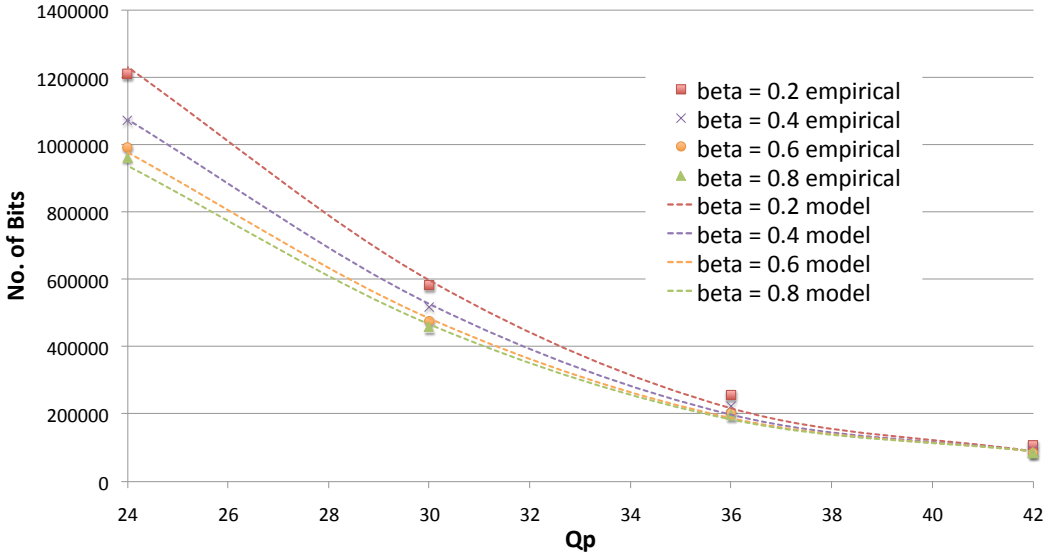



FIGURE 5.4: soccer: comparing actual number of encoder bits and fitted model.

where $\sigma_0$, $\sigma_1$ and $\sigma_2$ are model parameters and $R$ is the number of bits per GOP.

### 5.3.3 Complexity Model

Modeling the power consumption or complexity of video encoders for the purpose of system optimization has been proposed in [85] [53] and [91]. On a given computing platform, the encoding time gives a direct indication of the complexity of the encoder and also the power consumption of the encoder. When an encoder is operating under power constraint, it may not be able to encode at the speed required by the application. Depending on the platform the encoder is operating on, platform-specific power consumption model that relates program run-time and power consumption may be derived [53] [91]. The work in [85] measured the power consumption of an encoder, whose power consumption is controlled by the proportion of INTRA MBs, with an oscilloscope.

In our complexity scalable encoder, the encoding time varies near-linearly with control parameter $\beta$. When $\beta$ is zero, all MBs will be either SKIP or INTRA, so it is not surprising that the encoding time-$\beta$ curves of different $Q_p$ and of different sequences intersects the vertical axis at nearly the same point. Hence, we model the encoding time-$\beta$ relationship as

$$t = \mu_1 \beta + \mu_0, \tag{5.5}$$

where $t$ is the encoding time of one GOP in $ms$ and $\mu_1$ is the model parameter and $\mu_0$ is the vertical intersect. When the encoder is operating on a different computing platform, the intersect can be estimated from the time required to carry out R-D optimization only on the `INTRA` and `SKIP` mode. We assumed a linear relationship between the power consumption of

the video encoder and the encoding time: $P_{enc} = ct$ and vary the scaling factor $c$ to simulate encoder implemenation of different power consumption relative to the transmitter.

## 5.4 Complexity-Rate-Distortion Optimization

Instead of optimizing the distortion performance of the video encoding within only a rate constraint, we introduce a complexity constraint (in form of a time constraint) in our optimization. On a given computing platform, when the time constraint decreases (the encoder has to encoder faster), the encoder has to become less complex. When the power supply to the encoder decreases, the complexity of the encoder has to be decreased to maintain the same encoding time. The introduction of constraint is necessary if a real-time encoder is required but computational resource is inadequate.

Taking into account both rate and time constraints, we formulate the encoder optimization problem as:

$$\max_{\{q,\beta\}} D_{PSNR}(Q_p, \beta) = -0.6Q_p + \gamma_D \tag{5.6}$$

subject to $R < R_0$, where $R_0$ is the rate constraint and $R(Q_p, \beta) = (\sigma_2\beta^2 + \sigma_1\beta + \sigma_0)(Q_p - Q'_p)^2 + r'$, and $t < t_0$, where $t_0$ is the time constraint and $t = \mu_1\beta + \mu_0$. The output parameters, $\beta*$ and $Q_p*$ should maximize the fidelity measure (PSNR) while keeping the encoder within rate and complexity constraints.

Although the behaviour of the encoder can be described with simple models, the different model parameters are often sequence specific and may also vary throughout a sequence. Since the success of the optimization process is dependent on the accuracy of the models, appropriate model updates have to be carried out as encoding proceeds to ensure that the models track the changing characteristics of the encoded sequence.

One possible method is to encode each frame repeatedly until there is enough data to update all model parameters through linear regression. This may not be possible for an encoder operating under power constraint. Instead, we observe how the model parameters vary across sequences and design a simple update scheme. For models (5.3) and (5.4), we update the terms $\gamma_D$ and $\sigma_0$ after the encoding of each GOP, satisfied that this one step update is sufficient to track the changing characteristics of encoded sequences. To improve the accuracy of the model, $r'$ can also be updated periodically with a low complexity encoding of the input GOP (by choosing a small $\beta$) before the actually encoding of the GOP. The $\beta$ and $Q_p$ used for each frame can then be adjusted to track changing source characteristics or overcome model inaccuracies (Algorithm 4).

Joint rate and complexity control is not an easy problem as the chosen complexity control parameter also affects the rate. This problem is not unlike the problem of joint optimization

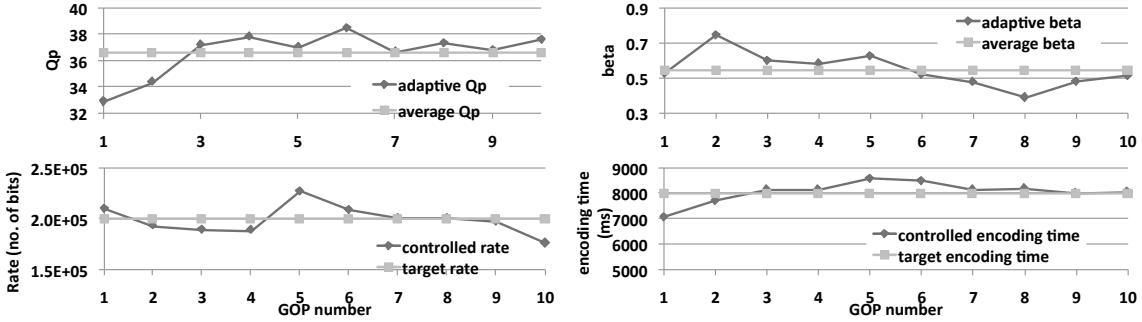



FIGURE 5.5: Crew: adapting $\beta$ and $Q_p$ to control rate and encoding time.

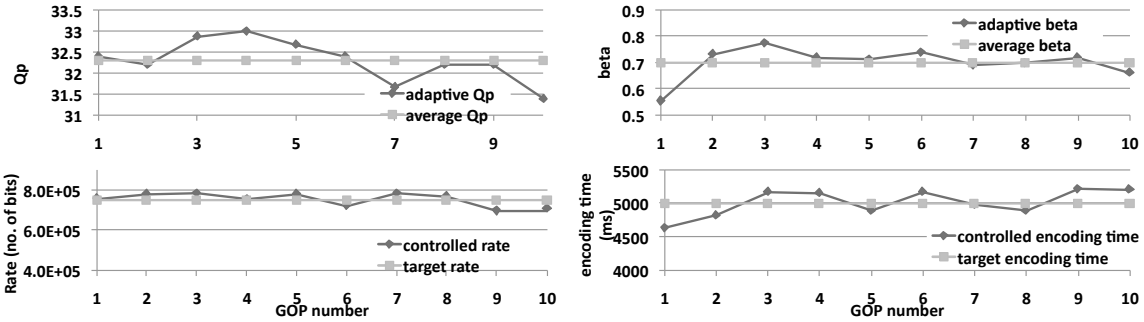



FIGURE 5.6: Mobile: adapting $\beta$ and $Q_p$ to control rate and encoding time.

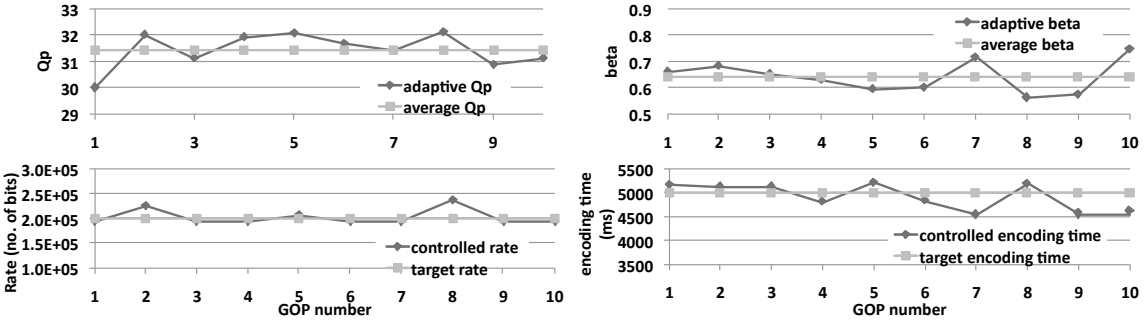



FIGURE 5.7: Silent: adapting $\beta$ and $Q_p$ to control rate and encoding time.

of vector dimensions and bit allocation for vector transform quantization in [93] which is solved with an iterative algorithm.

Figs. 5.5-5.7 shows how the complexity scalable encoder can be used to achieve simultaneous rate and complexity control.

## 5.5 Joint Video Encoder and Transmitter Power Allocation

In Section 5.2, we showed how joint source encoder and transmitter power allocation can be useful during the transform coding and transmission of a Gauss-Markov source. In this

section, we use the complexity-rate-distortion model of a complexity scalable video encode described earlier to demonstrate the value of joint encoder-transmitter power allocation.

Using the same assumption about the wireless channel as Section 5.2, we pose the following constrained optimization problem

$$\max_{P_{enc},P_t,R} D_{PSNR}(P_{enc}, R) \tag{5.7}$$

subject to

$$P_t + P_{enc} < P_{lim} \tag{5.8}$$

and

$$R_s < rC, \tag{5.9}$$

where $P_{enc}$ and $R$ are the power consumption and the rate of the source encoder and $P_{lim}$ is the power constraint. We consider the case where the video distortion is due to lossy compression in the source encoder, and the transmitter power and source rate is selected to ensure reliable delivery. We assume that some form of channel code is used to ensure reliable transmission. In the presence of channel coding, the available rate for the source encoder must be reduced to $R_s = rC$ where $r \in [0, 1]$ is the channel code rate.

Power invested in the video encoder can potentially improve coding performance (since more coding modes can be tested) while power allocated to the wireless transmitter will increase the channel capacity. Therefore, it is actually not obvious if a joint power allocation can bring significant improvement of system performance. By using an empirically-obtained model describing the complexity-rate-distortion behaviour of a complexity scalable video encoder, we can get a clear indication of the value of considering the source encoder and the wireless transmitter jointly during system optimization.

## 5.6 Experimental Results

### 5.6.1 Complexity Scalable Encoding

To demonstrate the complexity scalable nature of our encoding scheme, the rate-distortion performance of the encoder is measured ( Figs. 5.8 and 5.9 ) as the complexity control parameter $\beta$ is varied from 0.1 to 1.0 in steps of 0.1 (also shown in Chapter 3). It is clear that we were able to control the complexity-coding performance trade-off with a single parameter $\beta$.

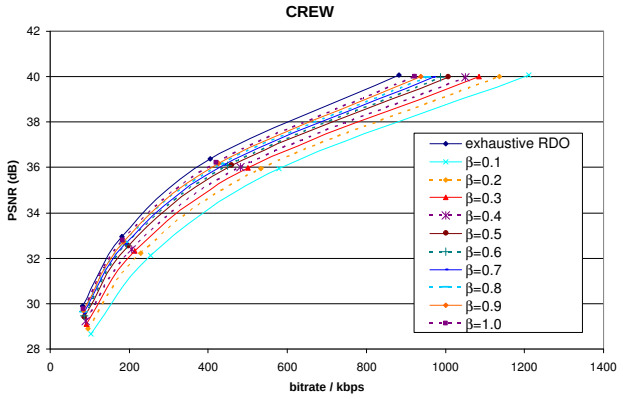



FIGURE 5.8: Complexity scalable encoding: rate-distortion performance against parameter $\beta$.

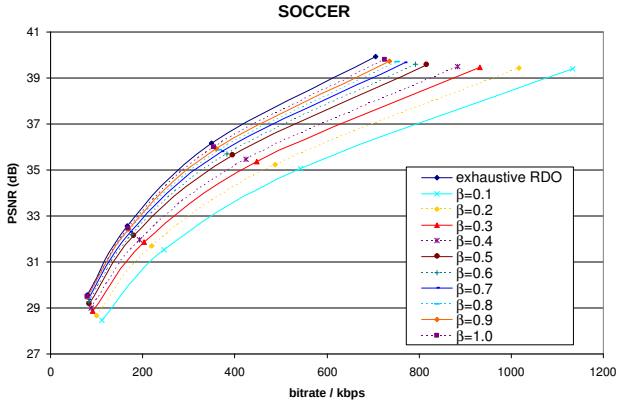



FIGURE 5.9: Complexity scalable encoding: rate-distortion performance against parameter $\beta$.

### 5.6.2 Joint Power Allocation

We use the models described earlier to illustrate the value of a power-scalable source encoder during overall system optimization.

Figs. 5.10-5.13 shows the simulation results for encoder of different scaling value $c$. In the simulation, we also used several encoders of fixed $\beta$, simulating encoders of fixed complexity-coding performance trade-off. Unsuprisingly, the best performance can be obtained when sytem allows optimal power allocation between the encoder and transmitter. When the power consumption of the encoder already surpasses the power limit of the system, the system will not be able to function. Therefore, using an encoder whose power consumption can be scaled back has the obvious advantage of increasing the power operating range of the system.

When system power is limited, over-investing in the source encoder can have serious impact on system performance as starving the transmitter of power reduces channel capacity and also the bit-rate of the encoded video; this is where a power scalable encoder is the most valuable.

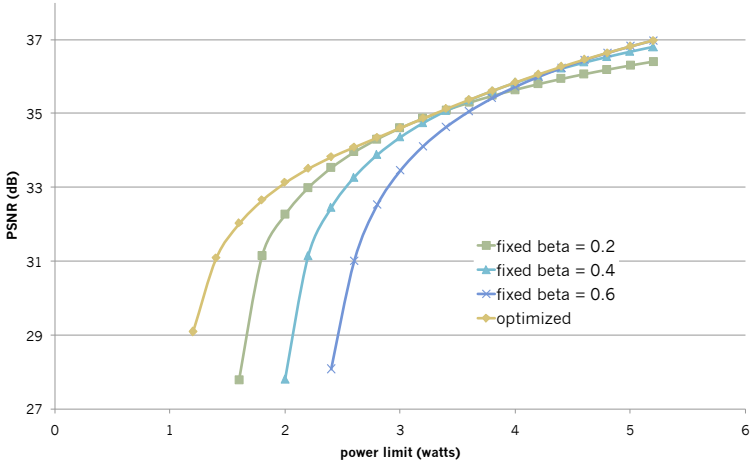



FIGURE 5.10: Operating under power constraint ($c = 5 \times 10^{-4}$): Comparing optimized $\beta$ selection with encoders of fixed $\beta$.

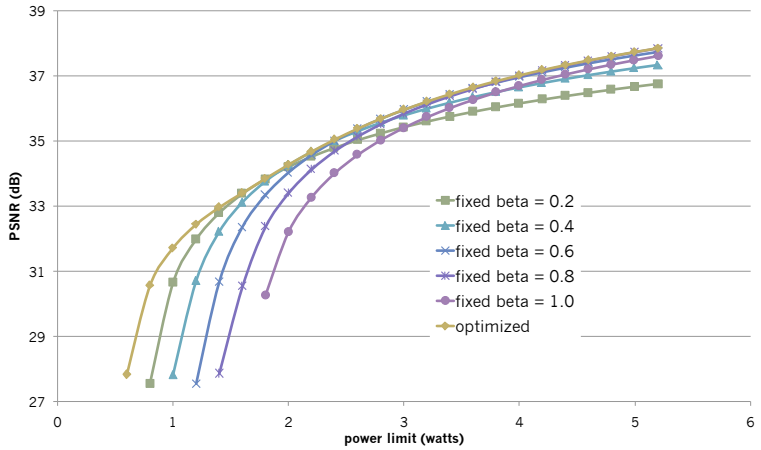



FIGURE 5.11: Operating under power constraint ($c = 2.5 \times 10^{-4}$): Comparing optimized $\beta$ selection with encoders of fixed $\beta$.

Using an encoder of low complexity-coding performance trade-off may alleviate the problem of poor transmitter-encoder power allocation at low power limit. However, this arrangement will cause the system performance to suffer as the power limit of the system increases and performance becomes constrained by the coding performance of the source encoder. Therefore, allowing flexible power allocation is useful, especially if varying system power limit is expected.

Comparing Figs. 5.10-5.13, it can be observed that the problem of over-investing in the source encoder is more serious if encoding is more power-consuming than transmission (higher $c$).

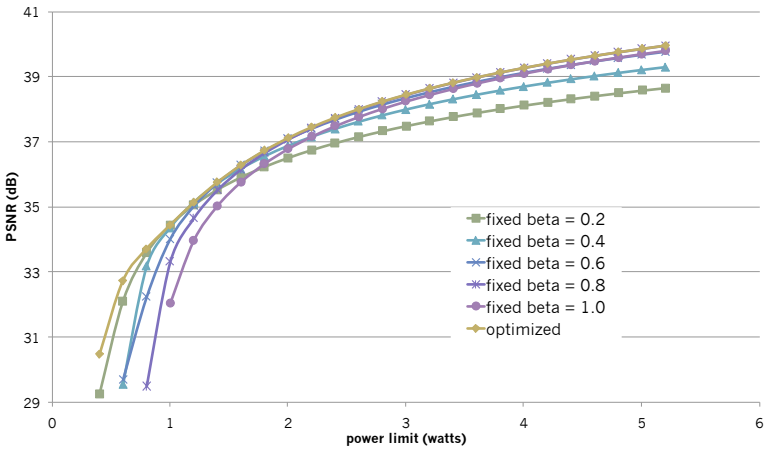



FIGURE 5.12: Operating under power constraint ($c = 1.25 \times 10^{-4}$): Comparing optimized $\beta$ selection with encoders of fixed $\beta$.

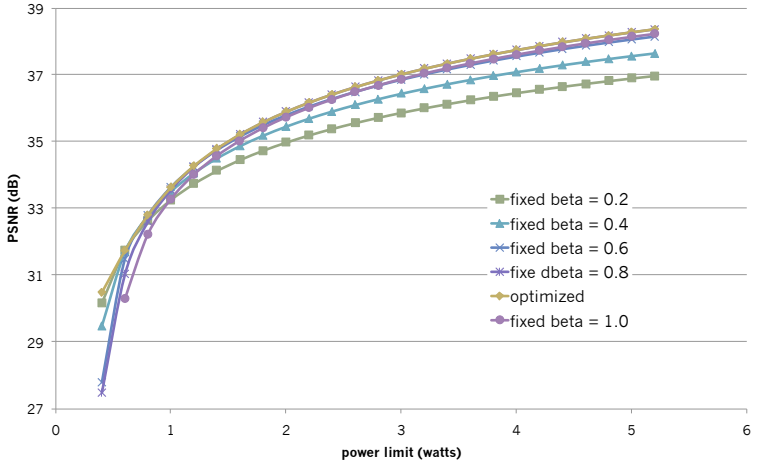



FIGURE 5.13: Operating under power constraint ($c = 0.0625 \times 10^{-4}$): Comparing optimized $\beta$ selection with encoders of fixed $\beta$.

## 5.7 Remarks

The design of a complexity scalable encoding scheme enables the optimization of a video encoder to be carried out with the availability of computational resource in consideration. This extension of traditional rate-distortion optimization is necessary when time or power constraints do not allow a video encoder to achieve rate-distortion optimized coding performance. In this chapter, we showed how the complexity scalable feature (Chapter 3) can be useful in the design of an encoder operating under both complexity and rate constraints. We also demonstrated that a complexity scalable encoder can allow flexible power allocation between encoder and transmitter, improving system performance under power constraint.

To make the best use of system resource, both effective complexity control (Chapter 3) and rate control (Chapter 4) is essential.

**Algorithm 4**: Complexity and Rate Control

**Input**: $T_{targ}^{input}$, target encoding time for each GOP.
**Input**: $R_{targ}^{input}$, target rate for each GOP.
**Input**: $Qp$, initial $Qp$
**Input**: $\beta$, initial $\beta$
**while** *(GOP*$\leftarrow$`Fetch_Next_GOP()`$\neq \emptyset$ **do**
  **Data**: $T_{GOP}$, encoding time for GOP.
  **Data**: $R_{GOP}$, rate for GOP.
  **Data**: $\beta_{av}$, average $\beta$ for GOP.
  **Data**: $Qp_{av}$, average $Qp$ for GOP.
  **Data**: $R_i$, rate for frame $i$.
  **Data**: $T_i$, target encoding time for frame $i$.
  $(R_i, T_i) \leftarrow$ `Encode`$(INTRA, Qp, \beta)$;
  $R_{targ} \leftarrow R_{targ}^{input} - R_i$;
  $T_{targ} \leftarrow T_{targ}^{input} - T_i$;
  $T_{GOP} \leftarrow T_{GOP} + T_i$;
  $R_{GOP} \leftarrow R_{GOP} + R_i$;
  $(Qp_{av}, \beta_{av}) \leftarrow$ `Compute_Average`$(Qp_{av}, \beta_{av}, Qp, \beta)$;
  **Data**: $GOPsize$, `Size` (GOP)-1.
  **while** *(FRAME*$\leftarrow$`Fetch_Next_FRAME()`$\neq \emptyset$ **do**
    **Data**: $Thres_{upper}$, upper threshold 1.2.
    **Data**: $Thres_{lower}$, lower threshold 0.8.
    $(R_i, T_i) \leftarrow$ `Encode`$(INTER, Qp, \beta)$;
    **if** $R_i \leq Thres_{lower} \times R_{targ}/GOPsize$ **then**
      $Qp \leftarrow Qp - 1$;
    **end**
    **if** $R_i \geq Thres_{upper} \times R_{targ}/GOPsize$ **then**
      $Qp \leftarrow Qp + 1$;
    **end**
    **if** $T_i \leq Thres_{lower} \times T_{targ}/GOPsize$ **then**
      $\beta \leftarrow \beta + 0.1$;
    **end**
    **if** $T_i \geq Thres_{upper} \times T_{targ}/GOPsize$ **then**
      $\beta \leftarrow \beta - 0.1$;
    **end**
    $T_{GOP} \leftarrow T_{GOP} + T_i$;
    $R_{GOP} \leftarrow R_{GOP} + R_i$;
    $(Qp_{av}, \beta_{av}) \leftarrow$ `Compute_Average`$(Qp_{av}, \beta_{av}, Qp, \beta)$;
  **end**
  $(\gamma_D, \sigma_0) \leftarrow$ `Update_Model`$(\beta_{av}, Qp_{av}, T_{GOP}, R_{GOP})$;
  $(\beta, Qp) \leftarrow$ `Compute_Parameter` $(\gamma_D, \sigma_0, T_{targ}^{input}, R_{targ}^{input})$;
**end**

# Chapter 6

# Designing a Network of Wireless Encoders

## 6.1 Introduction

Transmitting live videos from wireless devices is a challenging task due to the large amount of data that have to be transmitted and the delay sensitive nature of video delivery. From the perspective of an individual user, computational power or battery constraints may affect the system performance, given the power consuming nature of both video encoding and wireless transmission (Chapter 5). This motivated the authors of [85] to design algorithms to control the complexity of video encoders and model the characteristics of the resulting encoders before using the models to formulate constrained optimization problems that jointly optimize both video encoders and wireless transmitter. Other studies that analyzed the joint control of encoder and transmitter power during video or image transmission include [90] and [94].

The task of video delivery becomes more difficult when a group of encoders are sending video data over the same wireless network simultaneously. Not only is the performance of the network affected by the power constraints of individual users, the bandwidth and the amount of interference present in the network also become important factors. Transmitter power control is a key component of radio resource manangement. In a CDMA-like wireless network, each user spreads its signal over the entire bandwidth such that for any particular user, other users' signals appear as pseudo white noise. Therefore, the action of any individual user is likely to have an effect on the performance of other users. Hence, coordinating the transmitter power of the all participating users is an important step towards improving the performance of a wireless network. Goodman et al [95][96] proposed a power control scheme in which the base station broadcasts the value of a common signal to interference noise ratio that optimizes network performance. Video related work in [86] and [87] jointly optimized

source coding and transmission power under bandwidth constraint but did not consider the possibility of encoder power control.

In this work, we are primarily concerned with the behaviour of a group of power constrained encoders operating in the same wireless network. Depending on the interference caused by other users in the network, each player maximizes its individual performance defined by a utility function. The game settles at an equilibrium if one exists. In game theory, an equilibrium at which no player can increase his utility through unilateral action is a Nash equilibrium; a Nash equilibrium provides a stable point at which system performance can be studied. However, when all players maximize their utility in an independent, distributed fashion, the equilibrium may not be the best operating point. Saraydar et al demonstrated in their work [97] that pricing the system resource is a useful tool for achieving a more pareto efficient equilibrium. When a pricing mechanism is introduced in our system, power consumption of individual user can be significantly reduced. Responding to the capacity of the channel being affected by interference from other users, an encoder that can jointly control source rate and transmission power can reduce the source rate to ensure that rate of the encoded data remains within capacity. Using an empirically-derived complexity-rate-distortion model of a complexity scalable video encoder, we study the non-cooperative optimization of wireless video encoders operating in the presence of other transmitting encoders. Since the PSNR is a metric widely used to measure the fidelity of decoded pictures, it is ideal as a utility function to measure the level of satisfaction enjoyed by each user.

The joint optimization of wireless video encoders sharing a network was also studied in [98]. In that work, the authors minimized the total power consumption of all users in a wireless network subjected to a minimum video quality. The solution is a centralized one that simultaneously controls the operating parameters of each user in the network. In contrast, this work studies the performance and behaviour of a network of wireless encoders that carry out self optimization in a distributed fashion.

In the next section, we will describe the source encoder models that we used. In Section 6.2, we formulate the optimization that is carried out at each encoder. The transmission power control game is described in section 6.3 before we discuss our results in Section 6.4 and conclude in Section 6.5.

## 6.2 Optimized Power Allocation

In this section, we first describe the optimization framework that is used to determine encoder-transmitter power allocation for each encoder. With an empirically derived power-rate-distortion model, we then demonstrate that cross-layer (encoder/transmitter) power optimization is indeed a worthwhile step towards improving system performance.

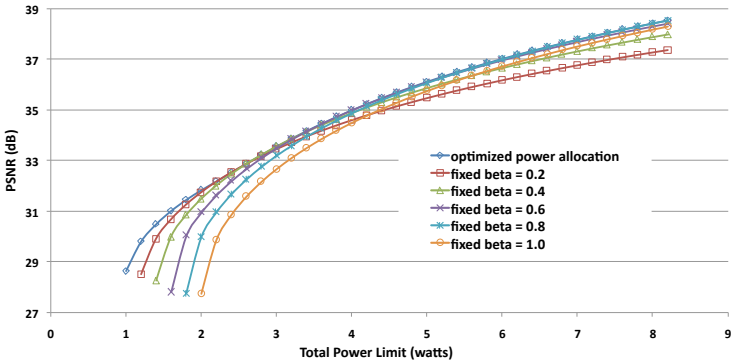



FIGURE 6.1: Comparing optimized performance of encoders with different complexity $\beta$ for $k = 1.25 \times 10^{-4}$.

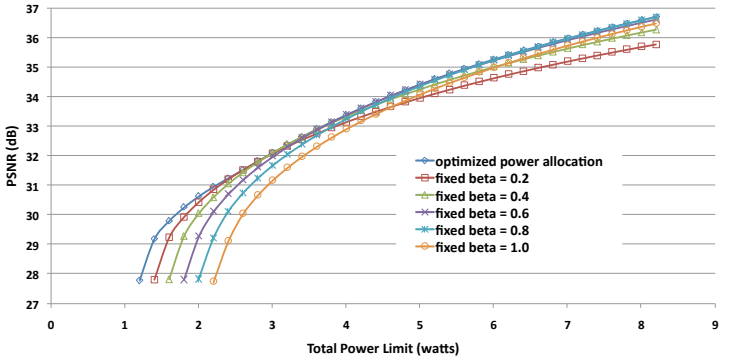



FIGURE 6.2: Comparing optimized performance of encoders of different complexity $\beta$ for $k = 2.5 \times 10^{-4}$.

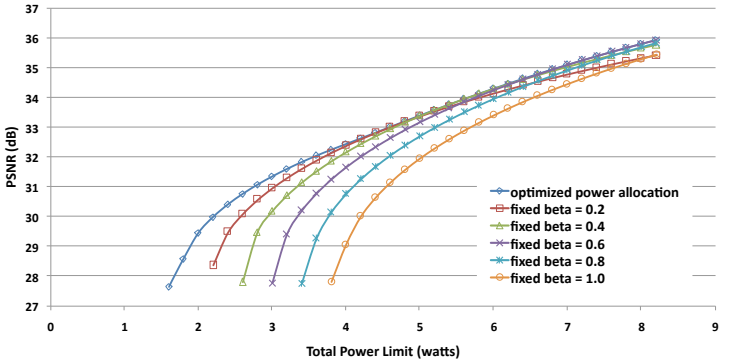



FIGURE 6.3: Comparing optimized performance of encoders of different complexity $\beta$ for $k = 5.0 \times 10^{-4}$.

In our power control game, each user attempts to optimize his performance independently by adjusting his encoder and transmitter power consumption within an overall power limit. We pose the following constrained optimization problem:

$$\min_{p_{enc},p_t,R_s} D(p_{enc}, p_t, R_s) \tag{6.1}$$

subject to

$$p_t + p_{enc} < p_{lim} \tag{6.2}$$

and

$$R_s < C, \tag{6.3}$$

where $p_{enc}$ and $R_s$ are the power consumption and the rate of the source encoder and $p_{lim}$ is the power constraint. The capacity $C$ of a band-limited ($W$ Hz) Gaussian channel with noise spectral density $N_0/2$ watts/Hz, power $p_t$ watts and attenuation $h$ is given by $C = Wlog(1 + \frac{hp_t}{N_0W})$ bits/s. We consider the case where the video distortion is due to lossy compression in the source encoder, and the transmitter power and source rate are selected to ensure reliable delivery. Assuming that some form of channel code is used to ensure reliable transmission, the available rate for the source encoder must be reduced to $R_s = rC$ where $r \in [0, 1]$ is the channel code rate. Therefore, the rate constraint on the source encoder becomes $R_s < rC$.

Using a power scalable encoder whose power consumption can be controlled by a single parameter $\beta$, and whose output rate can be controlled by the quantization parameter $Q_p$, the distortion and the rate of the encoder output are then determined by the $\beta$ and $Q_p$ and can be denoted as $D(\beta, Q_p)$ and $R_s(\beta, Q_p)$. If power consumption of the encoder, $p_{enc}(\beta)$ and that of the transmitter makes up the total power consumption, the problem becomes one of determining the optimal operating parameters of the encoder ($\beta^*$ and $Q_p^*$).

Power invested in the source encoder can potentially improve coding performance (more modes can be tested during rate-distortion optimization), while power allocated to the transmitter will increase channel capacity. Hence, it is not obvious if joint allocation will bring significant gains to system performance.

Figs 6.1-6.3 show the performance of a single user operating under fixed interference (from 8 other users on the network). The total power limit of the system is varied to obtain the three plots. The bandwidth of the wireless network is $1.5MHz$ and $N_0 = 5 \times 10^{-15}$ for all users. All transmitting encoders are $d = 800m$ away for the base station, resulting in the same attenuation ($h = 0.097/d^4$). If the power consumption of the source encoder already exceeds the total power constraint, then a viable system cannot be designed. This highlights an obvious advantage of a power scalable source encoder: it extends the range of total system power limit within which viable system can be designed. This advantage is more obvious when the value of $k$ is higher, that is, when the source encoder generally consumes more power than the wireless transmitter.

Even when the power consumption of the source encoder is within the total power limit, the allocation of power between the source encoder and the transmitter is important. When system power limit is low, over-investing in the encoder can have serious impact on system performance. This can be observed from the performance gap between encoders of high fixed $\beta$ (high fixed power consumption) and the case where encoder operating parameters (both $\beta$ and $Q_p$) are obtained to optimize system performance.

On the other hand, using encoders of low power consumption and poor coding performance (simulated with encoders with low fixed $\beta$) is also not an ideal solution. Though such a strategy may alleviate the problem of power deprivation of the transmitter at low overall power constraint, the coding performance of the source encoder becomes the limiting factor of overall system performance as total available power increases. This can be gathered from the simulation results (Figs 6.1-6.3); as total power limit increases, system using encoders of high power consumption and good coding performance (high fixed $\beta$) approaches the optimal performance while systems using low power encoders significantly underperform.

Since no encoder of fixed power consumption can ensure a good power allocation between source encoder and transmitter at different overall power limits, a power scalable encoder that can provide flexible encoder-transmitter power allocation is a useful tool.

## 6.3 Transmission Power Control Game

We consider the scenario where each wireless user encodes its video stream and competes with other users for the wireless medium to send the video data to the base station. When in operation, each user will have control over its $Q_p$, $\beta$ and transmitter power. We assume that all the videos that are being encoded follows the complexity-rate-distortion relationships described in the previous chapter. During the implementation of a practical system, model parameter estimation and updates will be necessary to track the varying characteristics of the source signal, this is a challenge faced by all model-based performance optimizing video delivery scheme.

Let $G = [M, P_j, u_j(\cdot)]$ denotes the transmission power control game (PCG) where $M = 1, ..., m$ is the index set for wireless encoders currently in the cell and $P_j$ is the strategy set. During the transmitter power control game, each encoder will select a transmitter power $p_j$ such that $p_j \in P_j$. The vector $\mathbf{p} = (p_1, ..., p_m)$ denotes the selected transmission power of all participating players. The utility for player $j$ at the end of the game, as a result of all selected strategies is $u_j(\mathbf{p})$.

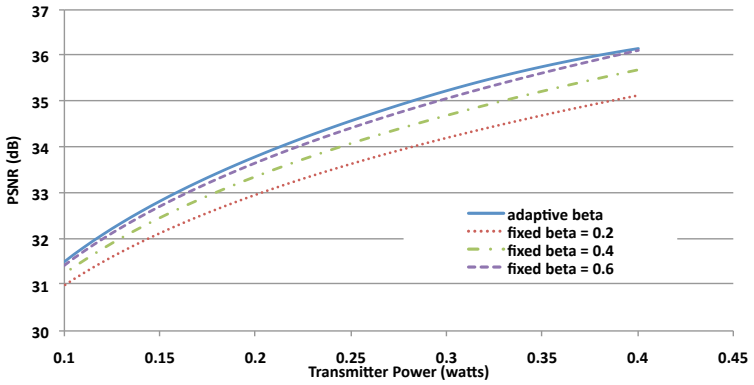



FIGURE 6.4: Utility functions of user transmission power for fixed interference. When the encoder is of fixed $\beta$, only $Q_p$ is a free parameter. When $\beta$ (power consumption) of the encoder can be varied, both $Q_p$ and $\beta$ are adjusted.

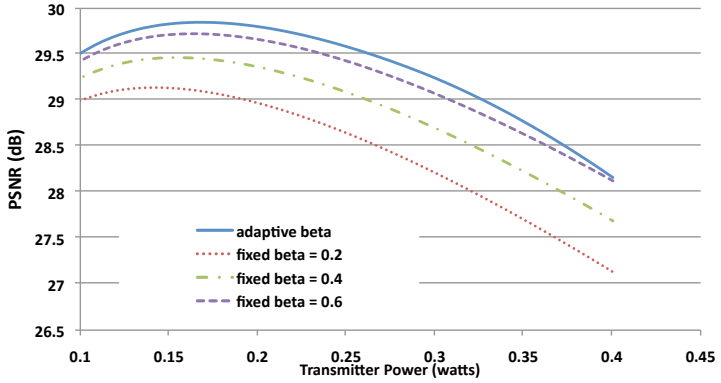



FIGURE 6.5: Utility functions ($u = PSNR - cp_t$) of user transmitted power for fixed interference ($c = 20$).

### 6.3.1 Utility Function

Utility refers to the level of satisfaction a player enjoys as a result of the actions of all players in the game. Since the PSNR is a widely used metric to measure the fidelity of decoded pictures, it is ideal as a utility function to measure the level of satisfaction enjoyed by each user. We express the utility user $j$ enjoys by choosing a strategy $p_j$ as $u_j(p_j, \mathbf{p_{-j}}) = D_{PSNR}(p_j, Q_{pj}, \beta_j)$, where $Q_{pj}$ and $\beta_j$ are encoder parameter controlling the quantization parameter and encoder power consumption respectively. The term $\mathbf{p_{-j}}$ refers to the power vector representing the transmitter power of all players except player $j$. It has been established that a Nash equilibrium exists when the utility function is quasi-concave[97]. The shape of the utility as function of user transmission power is shown in Fig. 6.4.

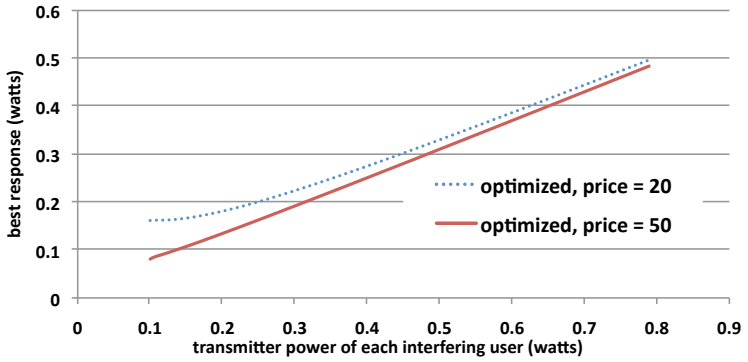



FIGURE 6.6: Response of player $j$ to 7 other players transmitting.

### 6.3.2 Response Function

Let player $j$'s best response to a given interference vector $p_{-j}$ be given by $r_j(p_{-j}) = \widetilde{p}_j$. We approximate the capacity available the player $j$ as $C = Wlog(1 + \frac{h_j p_j}{\sum_{i \neq j} h_i p_i + N})$bits/s where $h_j$ is the path gain from player $j$ to the base station and $N$ is the AWGN power at the receiver. To achieve optimal performance, player $j$ will wish to adopt strategy according to the operating conditions derived from the aforementioned optimization exercise (Section 6.2), resulting in the strategy $\widetilde{p}_j$. Since any deviation from $\widetilde{p}_j$ will result in sub-optimal performance for player $j$, $\widetilde{p}_j$ is the best response to power vector $\mathbf{p_{-j}}$.

A power vector $\mathbf{p}$ is a Nash equilibrium of the PCG $G = [M, P_j, u_j(\cdot)]$ if, for every $j \in M$, $u_j(p_j, \mathbf{p_{-j}}) \geq u_j(p'_j, \mathbf{p_{-j}})\ \forall p'_j \in P_j$. By definition, the Nash equilibrium has to satisfy $\mathbf{p} = \mathbf{r}(\mathbf{p})$ where $\mathbf{r}(\mathbf{p}) = (r_1(\mathbf{p}), ..., r_m(\mathbf{p}))$. In our simulation, we use the following iterative algorithm:

$$\mathbf{p}(t+1) = \mathbf{r}(\mathbf{p}(t)). \tag{6.4}$$

We observed that the response function often satisfies the property of a standard function [99](Fig. 6.6) and assumed that the iterative process converged at a unique equilibrium. For our experiment, we allow the system to converge before studying the performance at equilibrium. A function, $r(\mathbf{p})$ is said to be standard if it satisfies the following properties: positivity: $\mathbf{r}(\mathbf{p}) > 0$; monotonicity: if $\mathbf{p} \geq \mathbf{p}'$ then $\mathbf{r}(\mathbf{p}) \geq \mathbf{r}(\mathbf{p}')$; scalability: $\forall \mu > 1, \mu \mathbf{r}(\mathbf{p}) > \mathbf{r}(\mu \mathbf{p})$. As proven in [99], the iterative process converges to a unique point if $\mathbf{r}(\mathbf{p})$ standard.

### 6.3.3 Pricing

The work in [97] developed a non-cooperative game with pricing. The utility functions they used are in the form $u_j^c(\mathbf{p}) = u_j(\mathbf{p}) - f_j^c(p_j, \mathbf{p_{-j}})$, where $f_j^c : P \rightarrow \Re_+$ is the pricing function for player $j \in M$. In the same vein, we formulate our utility function with pricing as

$$u_j^c(p_j, \mathbf{p_{-j}}) = D_{PSNR}(p_j, Q_{pj}, \beta_j) - f_j^c(p_j, \mathbf{p_{-j}}), \tag{6.5}$$

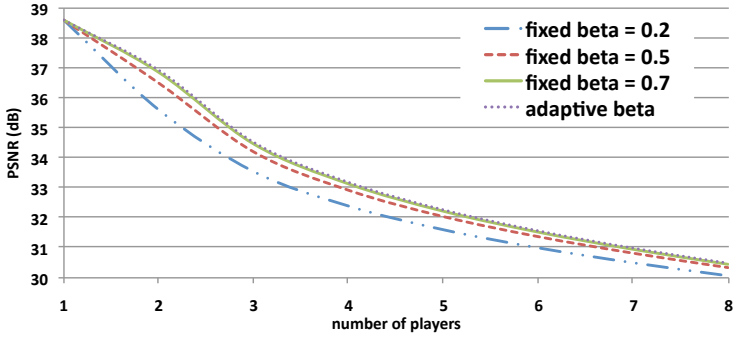



FIGURE 6.7: Performance degradation as more players join.

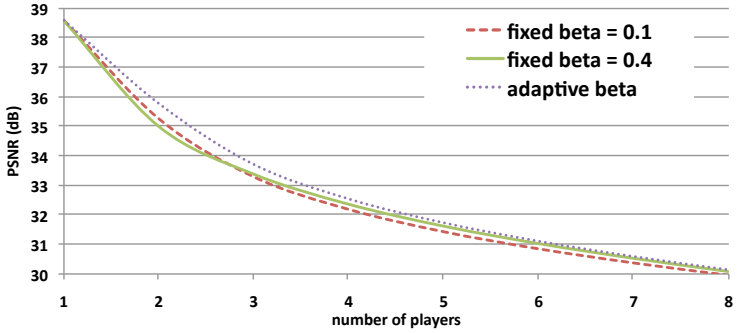



FIGURE 6.8: Performance degradation as more players join.

with

$$f_j^c(p_j, \mathbf{p}_{-\mathbf{j}}) = c_t p_j, \tag{6.6}$$

where $c_t$ places a unit cost on each player's transmission power.

The work in [98] determines the operating parameters of wireless video transmitters through a centralized optimization to minimize power consumption to maintain a minimum allowable distortion. Charging a unit cost on transmission power modifies the utility function (Fig. 6.5) and the optimization process of each individual user, causing all users to back off, reducing the interference of the wireless network. In this way, the aim of reducing transmission power while maintaining performance can be attained in a distributed fashion.

## 6.4 Numerical Results

For each of our simulations, we allow the system to iterate to an equilibrium before making comparisons. The following values were used for our system parameter: AWGN power at the receiver, $N = 5 \times 10^{-15}$watts; bandwidth, $W = 1.5 \times 10^6$Hz; code rate, $r = 0.5$; total power limit, $p_{lim} = 1.5$watts.

The universal frequency reuse nature of a CDMA-like network together with source rate control of the video encoders enable graceful performance degradation as more players join

the system. This can be observed in Fig. 6.7. In the simulation, $k = 0.125 \times 10^{-4}$ for all users and all users are $d = 800m$ away from the base station. Path gains are obtained using simple path loss model $h_j = Z/d_j^4$ where Z=0.097 is a constant. When encoder $\beta$ is fixed, only the transmitter power and the $Q_p$ is varied to optimized individual performance at each iteration. When $\beta$ can be varied, the power consumption of the encoder provides another degree of freedom.

Another set of simulation (Fig.6.8) shows the same experiment but with the overall power limit of each user reduced to 0.8 watts. In section 6.2, we have shown that it is possible to over-invest in the encoding processing when operating under power constraint and allocating insufficient power to maintain sufficient channel capacity. This results in significantly worse performance compared to an optimized cross-layer power allocation scheme. From the two sets of simulations, we can observe that the problem of over-investing in the source encoder becomes a non-issue in the setting of a network of wireless encoders; power allocated to the source encoder tends to improve performance as long as the power consumption of the encoder stays within the total power limit.

When designing a system of power constraint wireless encoders, a good rule of thumb to follow is to allocate most of the available power improving coding performance of the encoder. When all encoders adopt this strategy, the reduced interference and improved coding performance will likely lead to better individual performance for all players. Only with a power scalable source encoder and the flexibility to adjust power allocation between source encoder and transmitter can the system ensure such allocation with varying power limit.

Fig.6.10 shows how PSNR and transmission power vary with price. All users ($k = 0.2 \times 10^{-3}$) are at a distance $d = 800m$ away from base station. Increasing unit cost helps suppress the transmission power of each player, reducing their power consumption at a relatively small degradation of performance. Such a scheme is particularly suitable for video delivery as small degradations in objective measure (PSNR) are usually perceptually inconsequential. Charging a price on transmitter power causes a change in the utility functions on all users, resulting in a new equilibrium at which each user expends significantly less energy.

With $\beta$ set to 0.5 (fixed encoder power consumption), we track the PSNR and the transmitter power of each player. We consider eight players located at $d$=[800, 820, 840, 860, 880, 900, 920, 940]m from the base station. Pricing appears to help the system attain a fairer equilibrium, with smaller performance gap between players nearest and furthest the base station.

## 6.5 Remarks

We used an empirically-derived complexity-rate-distortion model to study the behaviour of a group of self-optimizing wireless video encoders. We observed how the universal frequency reuse nature of a CDMA-like network together with source rate control of the video encoders

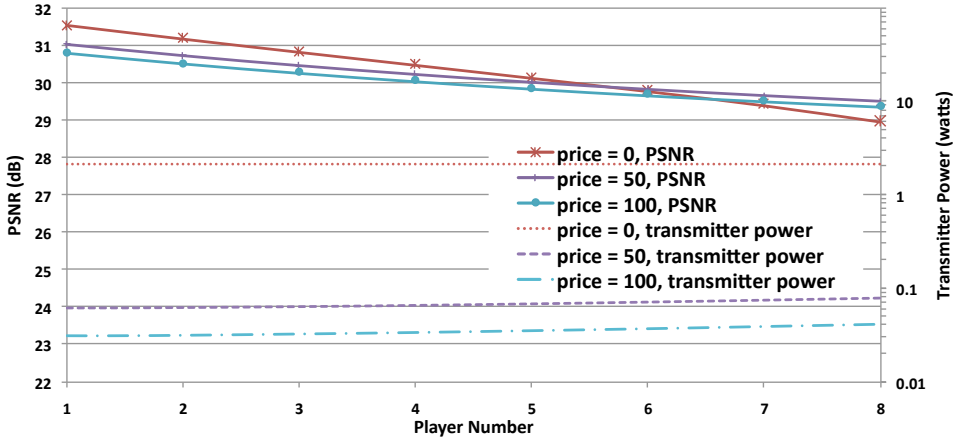



FIGURE 6.9: PSNR of each player at equilibrium (with different $\beta$ value simulating encoder of different power consumption).

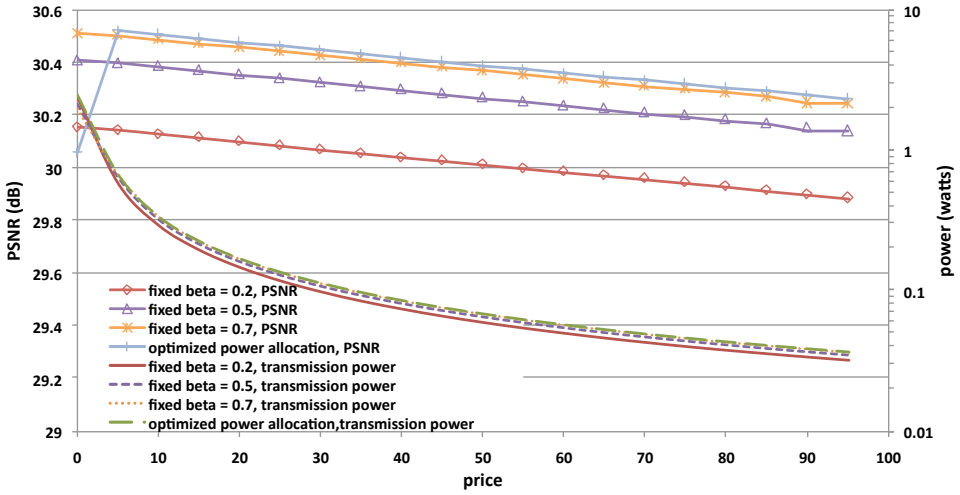



FIGURE 6.10: Equilibrium PSNR and transmitter power, $p_t$ of each player.

enable graceful performance degradation as number of players increased, the effectiveness of pricing at improving performance and the impact of different power allocation between encoder and transmitter. Our analysis on distributed power control is useful for the design of a network of wireless video encoders.

# Chapter 7

# Conclusions

## 7.1 Contributions

In this project, we first developed algorithms for video encoder complexity reduction, complexity control and rate control.

Complexity reduction: Making use of inter-layer correlation of coding modes and with cross-layer considerations during the encoding of the base layer, we were able to achieve significant complexity reduction during the encoding of videos with several quality layers. By taking advantage of the regularity of motion across several frames, we also designed a fast motion estimation scheme that is useful for encoding a group of hierarchically-arranged video frames.

Complexity control: When power consumption of the video encoder has to be reduced beyond what is possible with fast rate-distortion or reduced complexity motion estimation algorithms, we proposed a complexity scalable video encoding scheme that helps achieve a complexity-coding performance trade-off in video encoders. With this encoding scheme, power adaptive video encoders can be designed. A single implementation of the video encoder can also be deployed on different platforms with different computing capability.

Rate Control: With models that track the variation of both the header and texture bits with quantization parameter, we resolve the RDO/rate control conundrum by considering $Q_p$ for quantizing transform coefficients and the $\lambda$-determining $Q_p$ as separate parameters that are obtained from a constrained optimization procedure. Two one-step model update processes help the rate control scheme track the changing characteristics after the encoding of each basic unit. Since the updates only require information from the previously encoded basic unit, they provide fast approximations for the proposed rate control scheme to make $Q_p$ adjustments, resulting in tight rate control.

Effective video encoder control ensures that performance of a video delivery system can be optimized under system limitation. Other than being useful tools on their own, these algorithms are also useful in the study of system performance under different constraints.

Effective rate and power control are especially important for the design of power constraint real-time wireless video encoders. A complexity scalable encoding scheme that allows the control of complexity-coding performance trade-off enables the overall power optimization of a video encoding and delivery scheme. With an empirically derived complexity-rate-distortion model, we show the benefits of cross-layer optimization for power allocation for a single wireless video transmitter. We also described a framework that will help in the design of a power-efficient network of wireless encoders.

## 7.2 Future Work

Chapters 5 and 6 showed that the ideal of optimal cross-layer power allocation is a worthwhile goal during system design. To help achieve the goal, an encoder capable of joint rate and complexity (power) control can be built with methods described in Chapters 3 and 4.

For more practically relevant experiments, actual power consumption of a particular set of video encoder implementation and wireless transmitter can be measured. Dynamic voltage scaling can be used to control system power consumption. When working in conjunction with the aforementioned joint rate and complexity control-capable encoder, the effect of flexible encoder-transmitter power allocation can be studied.

To enjoy the potential benefits of optimized power allocation between encoder and transmitter, a predictive control model that can be updated to track the characteristics of the encoded video sequence is also required.

Considering errors introduced during transmission and the power consumption and performance of the channel encoder can also add another dimension to the analysis.

# Bibliography


[1] Y. H. Tan, Z. G. Li, K. P. Lim, and S. Rahardja, "Simplified motion refined scheme for fine granularity scalability," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, pp. 1212–1222, Sep. 2008.

[2] W. S. Lee, Y. H. Tan, J. Y. Tham, K. H. Goh, and D. Wu, "Lacing: An improved motion estimation framework for scalable video coding," *Proc. 16th ACM International Conference on Multimedia*, pp. 769–772, Nov. 2008.

[3] Y. H. Tan, Z. G. Li, and S. Rahardja, "Accurate H.264 rate control with new rate-distortion models," *Proc. IEEE Region 10 Conference*, Nov. 2009.

[4] Y. H. Tan, Z. G. Li, S. Rahardja, and K. M. Lye, "Video rate control with motion and texture rate-distortion models," *IEEE Trans. Circuits Syst. Video Technol.*, submitted.

[5] Y. H. Tan, W. S. Lee, J. Y. Tham, and S. Rahardja, "Complexity scalable rate-distortion optimization for H.264/AVC," *Proc. IEEE International Conference on Image Processing*, pp. 3397–3400, Nov. 2009.

[6] Y. H. Tan, W. S. Lee, J. Y. Tham, S. Rahardja, and K. M. Lye, "Complexity scalable h.264/avc encoding," *IEEE Trans. Circuits Syst. Video Technol.*, accepted.

[7] Y. H. Tan, W. S. Lee, and J. Y. Tham, "Complexity control and computational resource allocation during H.264/SVC encoding," *Proc. 17th ACM International Conference on Multimedia*, pp. 897–900, Nov. 2009.

[8] Y. H. Tan, W. S. Lee, J. Y. Tham, and S. Rahardja, "Complexity-rate-distortion optimization for real-time H.264/AVC encoding," *Proc. IEEE International Conference on Computer Communications and Networks*, pp. 1–6, 2009.

[9] ——, "Joint source coding-transmission power control for H.264 video delivery," *Proc. IEEE Region 10 Conference*, Nov. 2009.

[10] ——, "Complexity-rate-distortion analysis for constrained video encoding and transmission," *IEEE Trans. Multimedia*, submitted.

[11] ——, "Non-cooperative optimization of wireless video encoders," *Proc IEEE International Conference on Acoustics, Speech, and Signal Processing*, accepted.

[12] ——, "A resource allocation game for wireless video encoders," *IEEE Trans. Circuits Syst. Video Technol.*, submitted.

[13] 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, pp. 560–576, Jul. 2003.

[14] P. List, A. Joch, J. Lainema, G. Bjontegaard, and M. Karczewicz, "Adaptive deblocking filter," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, pp. 614–619, Jul. 2003.

[15] T. Wedi, "Motion compensation in H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, pp. 577–586, Jul. 2003.

[16] J. M. Boyce, "Weighted prediction in the H.264/MPEG AVC video coding standard," *Proc. International Symposium on Circuits and Systems*, pp. 789–792, 2004.

[17] A. M. Tourapis, F. Wu, and S. Li, "Direct mode coding for bipredictive slices in the H.264 standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, pp. 119–126, Jan. 2005.

[18] D. Marpe, H. Schwarz, and T. Wiegand, "Context-based adaptive binary arithmetic coding in the h.264/avc video compression standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, pp. 620–636, Jul. 2003.

[19] H. S. Malvar, A. Hallapuro, M. Karczewicz, and L. Kerofsky, "Low-complexity transform and quantization in h.264/avc," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, pp. 598–603, Jul. 2003.

[20] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, pp. 1103–1120, Sep. 2007.

[21] M. Wien, H. Schwarz, and T. Oelbaum, "Performance analysis of SVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, pp. 1194–1203, Sep. 2007.

[22] W. Li, "Overview of fine granularity scalability on MPEG-4 video standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 301–317, Mar. 2001.

[23] ——, "Bit-plane coding of DCT coefficients for fine granularity," *J ISO/IEC JTC1/SC29/WG11, MPEG98/M4025*, Oct. 1998.

[24] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. Journal*, vol. 27, pp. 379–423, 1948.

[25] H. Schwarz, D. Marpe, and T. Wiegand, "SNR-scalable extension of H.264/AVC," *IEEE International Conference on Image Processing*, vol. 5, pp. 3113 – 3116, 2004.

[26] F. Wu, S. Li, and Y.-Q. Zhang, "A framework for efficient progressive fine granularity scalable video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 332–344, Mar. 2001.

[27] X. Sun, Y. Zhou, Y. Wang, G. J. Sullivan, M. C. Lee, F. Wu, and S. Li, "Progressive fine granularity scalable (PFGS) video coding," *ISO/IEC JTC1/SC29/WG11 MPEG2004/M10569/S06*, Mar. 2004.

[28] Y. He, F. Wu, S. Li, Y. Zhong, and S. Yang, "H.26L-based fine granularity scalable video coding," *Proc. IEEE International Symposium on Circuits and Systems*, vol. 4, pp. 548–551, 2002.

[29] H. C. Huang, C. N. Wang, and T. Chiang, "A robust fine granularity scalability using trellis-based predictive leak," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, pp. 372–384, Jun. 2003.

[30] J. Reichel, M. Wien, and H. Schwarz, "Working draft 1.0 of 14496-10:200x/AMD 1 scalable video coding," *ISO/IEC JTC1/SC29/WG11 MPEG2005/N6901*, Jan. 2005.

[31] Z. G. Li, Y. Liu, and Y. C. Soh, "A novel SNR refinement scheme for scalable video coding," *IEEE International Conference on Image Processing*, vol. 3, pp. 644–647, 2005.

[32] M. Winken, H. Schwarz, D. Marpe, and T. Wiegand, "Adaptive motion refinement for FGS slices," *Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6 Bangkok, Thailand*, Jan. 2006.

[33] Z. G. Li, S. Rahardja, and H. Sun, "Implicit bit allocation for combined coarse granular scalability and spatial scalability," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, pp. 1449–1459, Dec. 2006.

[34] D. Wu, F. Pan, K. P. Lim, S. Wu, Z. G. Li, X. Lin, S. Rahardja, and C. C. Ko, "Fast intermode decision in H.264/AVC video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, pp. 813–822, Jul. 2005.

[35] 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, pp. 560–576, Jul. 2003.

[36] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. J. Sullivan, "Rate-constrained coder control and comparison of video coding standards," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, pp. 688–703, Jul. 2003.

[37] H. Schwarz, D. Marpe, and T. Wiegand, "Inter-layer prediction of motion and residual data," *ISO/IEC JTC1/SC 29/WG11/M11043*, Jul. 2004.

[38] K. Ugur, J. Lainema, A. Hallapuro, and M. Gabbouj, "Generating H.264/AVC compliant bitstreams for lightweight decoding operation suitable for mobile multimedia systems," *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, May 2006.

[39] H. Li and Z. G. Li, "Adaptive decoder complexity reduction for coarse granular scalability," *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, Apr. 2007.

[40] T. Koga, K. Ilinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing," *Proc. Nat. Telecomm. Conf.*, pp. G5.3.1–G5.3.5, Nov. 1981.

[41] J. R. Jain and A.K.Jain, "Displacement measurement and its application in interframe image coding," *IEEE Trans. Commun.*, vol. 29(12), pp. 1799–1808, 1981.

[42] J. Y. Tham, S. Ranganath, M. Ranganath, and A. A. Kassim, "A novel unrestricted center-biased diamond search for block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8(4), pp. 369–377, Aug. 1998.

[43] H. Schwarz, D. Marpe, and T. Wiegand, "Hierarchical B pictures," *Joint Video Team, Doc. JVT-PO14, Poznan, Poland*, Jul. 2005.

[44] "ITU-T rec. h.264 | iso/iec 14496-10. advanced video coding for generic audio-visual services," *Joint Video Team, Doc. JVT-V202*, 2005.

[45] T. Wiegand, G. J. Sullivan, J. Reichel, H. Schwarz, and M. Wien, "Joint scalable video model JSVM-9," *Joint Video Team, Doc. JVT-V202*, Jan. 2007.

[46] "Jsvm software. available from cvs repository." *:pserver:jvtuser@garcon.ient.rwth-aachen.de:/cvs/*.

[47] P. Yin, H.-Y. Tourapis, A. Tourapis, and J. Boyce, "Fast mode decision and motion estimation for JVT/H.264," *Proc. International Conference on Image Processing*, vol. 3, pp. 853–856, Sep. 2003.

[48] Y.-C. Lin, T. Fink, and E. Bellers, "Fast mode decision for H.264 based on rate-distortion cost estimation," *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1, pp. 1137–1140, Apr. 2007.

[49] I. Ahmad, W. G. Zheng, J. C. Luo, and M. Liou, "A fast adaptive motion estimation algorithm," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16(3), pp. 420–438, Mar. 2006.

[50] H.-Y. Cheong and A. Tourapis, "Fast motion estimation within the H.264 codec," *Proc. International Conference on Multimedia and Expo*, vol. 3, pp. 517–520, Jul. 2003.

[51] Z. Chen, J. Xu, Y. He, and J. Zheng, "Fast integer-pel and fractional-pel motion estimation for H.264/AVC," *Journal Visual Communication and Image Representation*, vol. 17(2), pp. 264–290, Apr. 2006.

[52] W. P. Burleson, P. Jain, and S. Venkatraman, "Dynamically parameterized architecture for power-aware video coding: Motion estimation and DCT," *Proc. 2nd USF Int. Workshop Digital and Computational Video*, pp. 8–12, 2001.

[53] 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.*, vol. 15, pp. 645–658, May 2005.

[54] I. Ismaeil, A. Docef, F. Kossentini, and R. Ward, "A computation-distortion optimized framework for efficient DCT-based video coding," *IEEE Trans. Multimedia*, vol. 3, pp. 298–310, Sep. 2001.

[55] K. Lengwehasatit and A. Ortega, "Computationally scalable partial distance based fast search motion estimation," *Proc. International Conference on Image Processing*, vol. 1, pp. 824–827, Sep. 2000.

[56] C. Kim, J. Xin, V. Anthony, and K. C.-C. Jay, "Complexity scalable motion estimation for H.264/AVC," *Visual communications and image processing*, vol. 6077(2), p. 60770, Jan. 2006.

[57] C. S. Kannangara, I. E. Richardson, and A. Miller, "Computational complexity management of a real-time H.264/AVC encoder," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, pp. 1191–1200, Sep. 2008.

[58] E. Akyol, D. Mukherjee, and Y. Liu, "Complexity control for real-time video coding," *Proc. International Conference on Image Processing*, vol. 1, pp. 77–80, Sep. 200.

[59] L. Su, Y. Lu, F. Wu, S. Li, and W. Gao, "Real-time video coding under power constraint based on H.264 codec," *SPIE Visual Communications and Image Processing*, vol. 6508, pp. 1–12, Feb. 2007.

[60] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Process. Mag.*, vol. 15, pp. 74–90, Nov. 1998.

[61] A. Ortega and K. Ramchandran, "Rate-distortion methods for image and video compression," *IEEE Signal Process. Mag.*, vol. 15, pp. 23–50, Nov. 1998.

[62] G. Amit and A. Pinhas, "Real-time H.264 encoding by thread-level parallelism: Gains and pitfalls," *Proc. of the 17th IASTED International Conference on Parallel and Distributed Computing and Systems*, 2005.

[63] J. Reichel, M. Wien, and H. Schwarz, "Working draft 1.0 of 14496-10:200x/amd 1 scalable video coding," *ISO/IEC JTC1/SC29/WG11 MPEG2005/N6901 Hong Kong*, Jan. 2005.

[64] "Information technology-coding of audio-visual objects-part 10: Advanced video coding. ISO/IEC FDIS 14496-10."

[65] H. Schwarz, D. Marpe, and T. Wiegand, "Inter-layer prediction of motion and residual data," *ISO/IEC JTC1/SC 29/WG11/M11043*, Jul. 2004.

[66] ——, "SNR-scalable extension of H.264/AVC," *Proc. IEEE International Conference on Image Processing*, pp. 3113–3116, Oct. 2004.

[67] G. Cheung and S. McCanne, "A framework for computation-memory algorithmic optimization for signal processing," *IEEE Trans. Multimedia*, vol. 5, pp. 174–185, Jun. 2003.

[68] V. Goyal and M. Vetterli, "Computation-distortion characteristics of block transform coding," *Proc. International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, pp. 2729–2732, Apr. 1997.

[69] A. P. Chandrakasan, D. C. Daly, D. F. Finchelstein, J. Kwong, Y. K. Ramadass, M. E. Sinangil, V. Sze, and N. Verma, "Technologies for ultra dynamic voltage scaling," *Proc. IEEE*, vol. 98(2), pp. 191–214, Feb. 2010.

[70] H. Schwarz, D. Marpe, and T. Wiegand, "Analysis of hierarchical B-pictures and MCTF," *Proc. IEEE Int. Conf. Multimedia Expo*, pp. 1929–1932, Jul. 2006.

[71] "Test model 5. [online]. available: http://www.mpeg.org/MPEG/MSSG/tm5/."

[72] J. Ribas-Corbera and S. Lei, "Rate control in DCT video coding for low delay communications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9(1), pp. 172–185, Feb. 1999.

[73] A. Vetro, H. Sun, and Y. Wang, "MPEG-4 rate control for multiple video objects," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9(1), pp. 186–199, Feb. 1999.

[74] T. Chiang and Y. Zhang, "A new rate control scheme using quadratic rate distortion model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7(1), pp. 246–250, Feb. 1997.

[75] Z. He, "A unified rate-distortion analysis frame work for transform coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11(12), pp. 1221–1236, Dec. 2001.

[76] W. Ding and B. Liu, "Rate control of MPEG video coding and recording by rate-quantization modeling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 12–19, Feb. 1996.

[77] K. Ramchandran, A. Ortega, and M. Vetterli, "Bit allocation for dependent quantization with applications to multiresolution and MPEG video coders," *IEEE Trans. Image Process.*, vol. 3, pp. 533–545, Sep. 1994.

[78] Z. G. Li, F. Pan, K. P. Lim, and S. Rahardja, "Adaptive rate control for H.264," *Proc. IEEE International Conference on Image Processing*, pp. 745–748, Oct. 2004.

[79] D.-K. Kwon, M.-Y. Shen, and C.-C. J. Kuo, "Rate control for H.264 video with enhanced rate and distortion models," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17(5), pp. 517–529, May 2007.

[80] E. Y. Lam and J. W. Goodman, "A mathematical analysis of the DCT coefficient distributions for images," *IEEE Trans. Image Process.*, vol. 9, pp. 1661–1666, Oct. 2000.

[81] F. Bellifemine, A. Capellino, A. Chimienti, R. Picco, and R. Ponti, "Statistical analysis of the 2D-DCT coefficients of differential signal for images," *Signal Processing: Image Commun.*, vol. 4, pp. 477–488, 1992.

[82] H.-M. Hang and J.-J. Chen, "Source model for transform video coder and its application - part 1: Fundamental theory," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 287–298, Apr. 1997.

[83] B. Tao, H. A. Peterson, and B. W. Dickinson, "A rate-quantization model for MPEG encoders," *Proc. IEEE International Conference on Image Processing*, pp. 338–341, Oct. 1997.

[84] Y. Liu, Z. G. Li, and Y. C. Soh, "A novel rate control scheme for low delay video communication of H.264/AVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, pp. 68–78, Jan. 2007.

[85] X. Lu, E. Erkip, Y. Wang, and D. Goodman, "Power efficient multimedia communication over wireless channels," *IEEE J. Sel. Areas Commun.*, vol. 21, pp. 1738–1751, Dec. 2003.

[86] Y. S. Chan and J. W. Modestino, "A joint source coding-power control approach for video transmission over CDMA networks," *IEEE J. Sel. Areas Commun.*, vol. 12(10), pp. 1516–1525, Dec. 2003.

[87] Y. Eisenberg, C. E. Luna, T. N. Pappas, R. Berry, and A. K. Katsaggelos, "Joint source coding and transmission power management for energy efficient wireless video communications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, pp. 411–424, Jun. 2002.

[88] Q. Zhang, Z. Ji, W. Zhu, and Y.-Q. Zhang, "Power-minimized bit allocation for video communication over wireless channels," *IEEE Trans. Circuits Syst. Video Technol.*, pp. 398–410, Jun. 2002.

[89] Z. Ji, Q. Zhang, W. Zhu, J. Lu, and Y.-Q. Zhang, "Joint power and source-channel coding for video communication over wireless networks," *Proc. IEEE Vehicular Technology Conf.*, pp. 1658–1662, Oct. 2001.

[90] T. Lan and A. H. Tewfik, "Power optimized mode selection for H.263 video coding and wireless communications," *Proc. International Conference on Image Processing*, vol. 2, pp. 113–117, 1998.

[91] Z. He and D. Wu, "Resource allocation and performance analysis of wireless video sensors," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, pp. 590–599, May 2006.

[92] C. E. Luna, Y. Eisenberg, R. Berry, T. N. Pappas, and A. K. Katsaggelos, "Joint source coding and data rate adaptation for energy efficient wireless video streaming," *IEEE J. Sel. Areas Commun.*, vol. 21(10), pp. 1710–1720, Dec. 2003.

[93] V. Cuperman, "Joint bit allocation and dimensions optimization for vector transform quantization," *IEEE Trans. Inf. Theory*, vol. 39, pp. 302–305, 1993.

[94] S. Appadwedula, M.Goel, N. R. Shanbhag, D. L. Jones, and K. Ramchandran, "Total system energy minimization for wireless image transmission," *J. VLSI Signal Process. Syst.*, vol. 1, pp. 99–117, Feb. 2001.

[95] D. Goodman and N. Mandayam, "Power control for wireless data," *IEEE Personal Communications*, vol. 7, pp. 48–54, 2000.

[96] ——, "Network assisted power control for wireless data," *Mobile Networks and Applications*, vol. 2, pp. 1022–1026, 2001.

[97] C. U. Saraydar, N. B. Mandayam, and D. J. Goodman, "Efficient power control via pricing in wireless data networks," *IEEE Trans. Commun.*, pp. 291–303, Feb. 2002.

[98] X. Lu, Y. Wang, E. Erkip, and D. Goodman, "Total power minimization for multiuser video communications over CDMA networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17(6), pp. 1051–8215, Jun. 2007.

[99] R. D. Yates, "A framework for uplink power control in cellular radio systems," *IEEE J. Sel. Areas Commun.*, pp. 1341–1347, Sep. 1995.