IMAGE AND VIDEO PROCESSING FOR DENOISING, CODING AND CONTENT PROTECTION

By

QIAN CHEN

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2011

I dedicate this dissertation to my beloved parents and my husband.

ACKNOWLEDGMENTS

In the first place, I would like to express my sincere gratitude to my advisor Prof. Dapeng Wu for his guidance and help in the development of my research. His constructive advice, highly responsible teaching, constant trust and willingness to provide funding play a significant role in the carrying out of this work.

Second, I would also like to thank Prof. Scott Banks, Prof. Andy Li, and Prof. Yijun Sun for serving on my dissertation committee and providing valuable suggestions on this dissertation. Meanwhile, I would like to express my special appreciation to Yunfei Zheng and Peng Yin for their valuable suggestions that help me to successfully complete my intern work in Technicolor Inc., which becomes part of the dissertation; and Jun Tian from Futurewei Inc., Zhifeng Chen from Interdigital LLC., for their kind guidance and useful advice in support of some of the work in the dissertation.

Last but not least, I owe my warmest thanks to my beloved parents and my dear husband, this work would not be possible without their continuous encouragement and support during the years of my PhD life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

IMAGE AND VIDEO PROCESSING FOR DENOISING, CODING AND CONTENT PROTECTION

By

Qian Chen

December 2011

Chair: Dapeng Wu
Major: Electrical and Computer Engineering

A typical digital video communication system consists of the following main parts: video capture, pre-processing, encoding, video security, video streaming over network, decoding and post-processing. In this dissertation, we address four parts of the system, denoising in pre-processing, coding efficiency improvement in encoding, content protection in security and delay requirement in video streaming.

In the first part, we propose a novel transform-domain based image denoising approach. It groups similar 2D fragments of image block into 3D data arrays to enhance the sparse representation of the signal in transform domain, so that the noise can be well separated by shrinking the coefficients. The proposed bounded BM3D turns out to be particularly effective for highly noisy images.

In the second part, we focus on the adaptive in-loop filter (ALF) module to improve the coding efficiency. We propose a classified quadtree-based adaptive in-loop filter (CQALF). The output pixels of deblocking filter are classified into two categories: those modified by the deblocking filter and those untouched. A filter will be designed for each category. By considering the impact of the deblocking filter when designing ALF, it overcomes over-filtering.

In the third part, we study the robust video hashing which applies to automatic content protection such as video identification and authentication. We propose to derive shot hash in a pair-wise manner. To utilize both temporal and spatial information,

2D-DCT is applied to each frame in a shot, quantize the DCT coefficient and record the temporal occurrence of the co-located coefficient. We select a pair of closet value, inverse transform to spatial domain and derive image hashes from two feature frames by Radial hASH.

In the last part, we emphasize the end-to-end delay in video communication system and propose a model to trade-off delay, rate and distortion for source coding in both real-time and delay insensitive applications. For real-time IPPPP coding mode, we verify the model accuracy and analyze the wireless channel transmission performance with varying encoding time through experiments. It is concluded that the proposed d-R-D model can accurately estimate the true coding distortion and rate, which is essential for further delay QoS control use.

CHAPTER 1
INTRODUCTION

1.1 Problem Statement

Multimedia technology, computer networks, and wireless communications have experienced massive growth and commercial success in recent years. As these technologies merge, multimedia communication, especially video communication, such as video conference, video surveillance, emergency response, tele-medicine, and multimedia systems in consumer electronics (smartphone, touchpad etc.), is expected to achieve unprecedented attention worldwide for the next few decades.

Therefore, technologies that can enhance the experience of visual quality and security of a video in a video communication system become more demanding. A typical video communication system consists of the following parts: video capture, pre-processing, video encoding, content security, video streaming over networks, video decoding and post-processing. Figure 1-1 shows a diagram of the system.

Images and videos captured by the device are usually subject to pre-processing techniques. Among those techniques, denoising is of particular interest and still poses a challenge. The huge amount of video data requires efficient compression to enable storage, transmission, and video on demand over networks. Hence the video coding beyond the state-of-art H.264/AVC is emerging to make higher compression efficiency possible. Meanwhile, though the computer and network interconnection makes the distribution of digital media fast and effortless, it poses great challenge to copyright protection. Robust video content protection is required so that the unauthorized duplication can be detected and the illegal distribution of the copyrighted work is prohibited.

1.1.1 Image and Video Denoising

The need for efficient image restoration methods has grown with the massive production of digital images and videos of all kinds, often taken in poor conditions. No

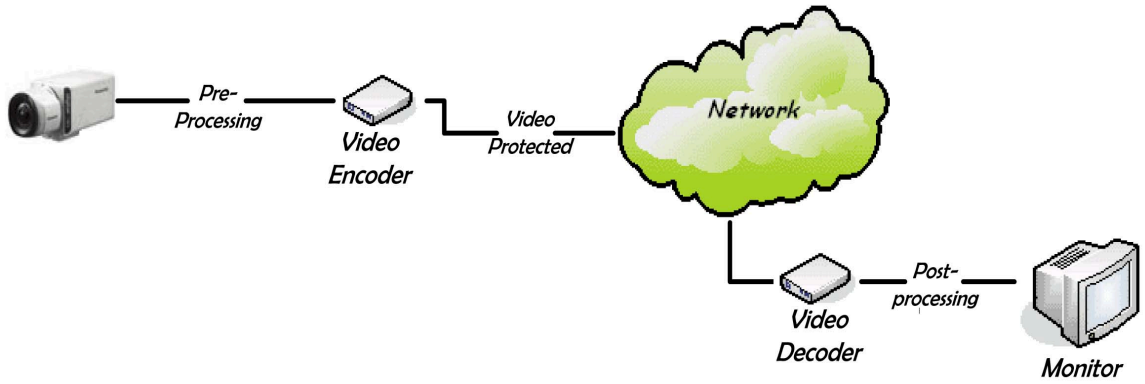


Figure 1-1. Diagram of a video communication system.

matter how good cameras are, an image improvement is always desirable to extend their range of action [1]. A digital image can be generally regarded as a matrix of grey-level or color values. In the case of a video, it becomes a 3D matrix with the third dimension being time. Therefore, image denoising methods can always apply to individual video frame for video denoising, making the two problems essentially one in theory. The two main limitations in digital image and video accuracy are blur and noise [1]. Blur is intrinsic to image acquisition systems, as digital images have a finite number of samples and must satisfy the Shannon-Nyquist sampling conditions [2]. The second main visual perturbation is noise.

In images captured by camera, each pixel value is the result of a light intensity measurement, usually made by a charge coupled device (CCD) matrix coupled with a light focusing system. Such a picture acquisition system can be modeled as an additive white noise as:

$$y(i) = x(i) + n(i) \tag{1–1}$$

where $i \in I$, $y(i)$ is the observed pixel value, $x(i)$ is the true value at pixel i, and $n(i)$ is the noise perturbation, and assumed to be independent random variables at different pixel positions.

The noise situation in videos is more complicated than in images. In addition to the additive white noise from the picture acquisition system, the artifacts become more

visually annoying and of more varieties in reconstructed videos after compression. The widely-used lossy video compression is based on a motion compensated hybrid of temporal DPCM and block DCT, including H.261, H.263, MPEG2, and H.264/AVC. Consequently, how to remove the coding artifacts produced in the coding process with moderate complexity is of equal importance, if not more emphasized, as the coding efficiency itself.

To acquire a better understanding of the motivation behind the second and third chapter of the dissertation, we will briefly review the coding artifacts that are mostly common in all lossy video codec.

1.1.1.1 Brief review of coding artifacts

1.Blocking Effect

We define the blocking effect as the discontinuities at the boundaries of adjacent blocks in a reconstructed frame. The cause of the blocking effect with respect to block-based coding is intuitively obvious, and has already been well documented[3][4]. In brief, due to the isolated nature in which individual blocks are coded, the level and characteristics of the coding error introduced into a block may differ from one block to another. This, in consequence, manifests as discontinuities between the boundaries of adjacent blocks. Examples of the blocking effect are shown in Figure 1-2 and Figure 1-3, which is enlarged parts of coding frame *mobile* and *Stefan*. We display blocking effect under different quantization steps, and also do comparison coded at the same QP before and after the deblocking filter in H.264. Clearly, the one coded with smaller QP has better quality, and using deblocking filter further improves the visual performance.

2. Blurring

Blurring manifests as a loss of spatial detail and a reduction in sharpness of edges in moderate to high spatial activity regions of frames, such as in roughly textured areas or around scene object edges. Figure 1-4 shows an example of the blurring effect in an enlarged part of frame *Bridge*. We display the blurring effect under different quantization.

Figure 1-2. Blocking artifacts in *Mobile* frame coded in H.264/AVC

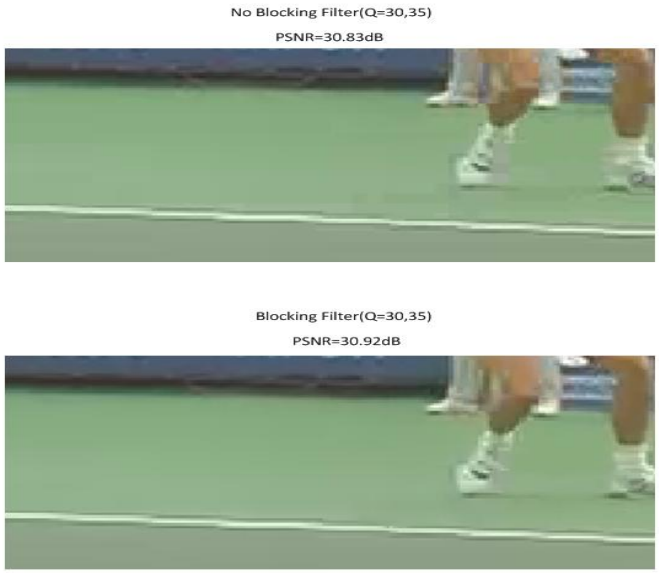


Figure 1-3. Blocking artifacts in *Stefan* frame coded in H.264/AVC

Figure 1-4. Blurring artifacts in *Bridge* frame coded in H.264/AVC

Clearly, the one coded with smaller QP has better visual quality. Note how the texture of the river becomes more and more visible from QP=40,45 to QP=20,25.

For intra coded macroblocks, blurring is directly related to the suppression of the higher order AC DCT coefficients through coarse quantization, leaving only the lower-order coefficients to represent the contents of a block; therefore, blurring can be directly associated with low-pass filtering. Similarly, in many respects, blurring, as a consequence of transform coding, can be considered as a specific case of the basis image effect where the prominent AC basis images after quantization are of a lower frequency, resulting in a reconstructed block with low spatial activity. For inter coded macroblocks, blurring is mainly a consequence of the use of a predicted macroblock with a lack of spatial detail.

3. Color Bleeding

The blurring of the luminance information results in the smoothing of spatial detail. The corresponding effect for the chrominance information results in a smearing of the color between areas of strongly contrasting chrominance. As with blurring, color

bleeding results from the quantization to zero of the higher-order AC coefficients, resulting in the representation of the chrominance components with only the lower frequency basis images. For chrominance edges of very high contrast, or where the quantization of the higher order AC coefficients does not result in their truncation, the color artifact corresponding to the ringing effect occurs.

It is interesting to note that strong chrominance edges are accompanied by strong luminance edges; however, the existence of a strong luminance edge does not necessarily coincide with a strong chrominance edge[5]. Therefore, color bleeding is not necessarily found at blurred edges in a reconstructed color frame.

4. Staircase Effect

The DCT basis images are not attuned to the representation of diagonal edges and features[6]. Consequently, more of the higher activity basis images are required to satisfactorily represent diagonal edges or significant diagonally oriented features. Due to the typically low magnitude of the higher-order basis images, coarse quantization results in their truncation to zero. The contribution originally made by the higher-order basis images in forming the diagonal edge is diminished, resulting in a reconstruction exhibiting only the characteristics of the lower frequency basis images, which are generally either horizontally or vertically oriented. So, for a block containing a diagonal edge angled towards the horizontal, coarse quantization will result in a reconstruction with a horizontal orientation, and vice versa for blocks angled towards the vertical.

The staircase effect is related to both the blocking and mosaic pattern effects in terms of the manifestation of discontinuities between adjacent blocks. When a diagonal edge is represented within a string of consecutive blocks, the consequence of coarse quantization is the reconstruction of the diagonal edge as a number of horizontal or vertical steps. These individual steps do not merge smoothly at a block's boundary with the continuance of the edge in adjacent blocks. A number of horizontally oriented examples of the staircase effect are shown in Figure 1-5 coded with different QP. Note

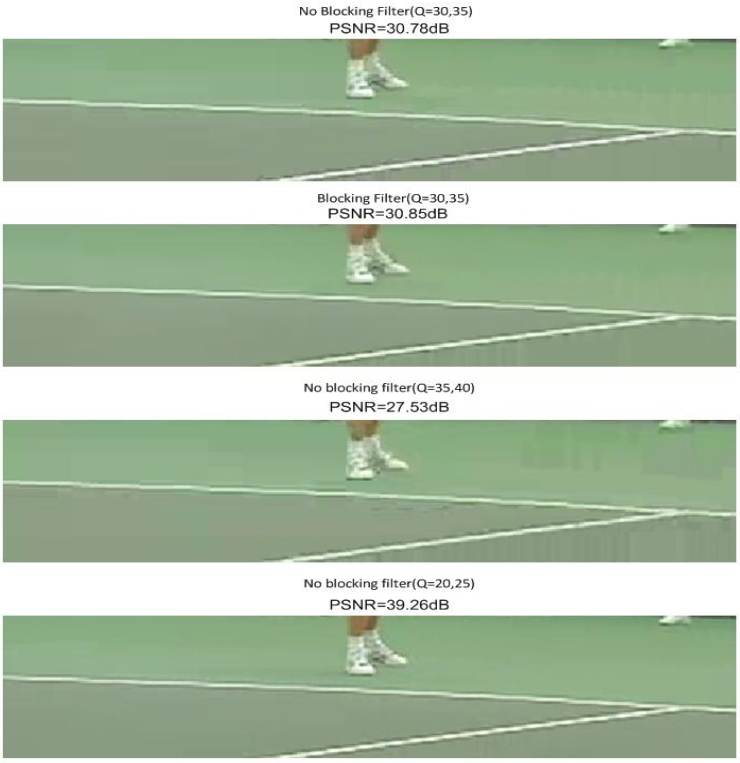


Figure 1-5. Staircase artifacts in *Stefan* frame coded in H.264/AVC

that the step-wise discontinuities occur at block boundaries in enlarged part of frame *Stefan*.

5. Ringing Effect

The ringing effect is most evident along high contrast edges in areas of generally smooth texture in the reconstruction, and appears as a shimmering or rippling outwards from the edge up to the encompassing block's boundary. The higher the contrast of the edge, the greater the level of the peaks and troughs of the rippling. Examples of this are shown in Figure 1-6, where it is most evident along the edge of the table-tennis table and the bottom of the player's arm. The generally smooth texture in the surrounding blocks results in a greater visibility of the ringing effect, where otherwise a masking of the ringing would be introduced.

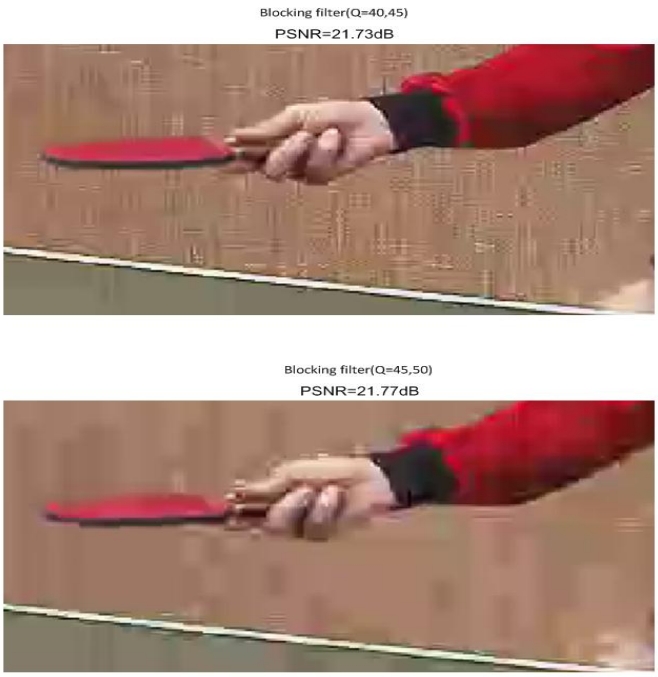


Figure 1-6. Ringing artifacts comparison before and after deblocking filter in *Table_tennis* frame coded in H.264/AVC

1.1.1.2 Research challenges

The search for efficient image denoising methods is still a valid challenge. All the current existing methods rely on some explicit or implicit assumptions about the true (noise-free) signal in order to separate it properly from the random noise [7]. In particular, the transform-domain denoising typically assumes that the signal is sparsely represented in the transform domain. The multiresolution transforms can achieve good sparsity for spatial details such as edges and singularities. A number of advanced denoising methods of this kind have been developed in [8][9][10][11]. However, the great variety of natural images makes it impossible for any fixed 2D transform to achieve good sparsity for all cases. The adaptive principal components of local image patches was proposed in [12] to overcome the drawbacks of standard orthogonal transforms. However, the computation of the correct PCA basis is essentially deteriorated by the presence of noise [7].

Recently, Dabov etc. proposed a novel image denoising strategy based on an enhanced sparse representation in transform domain, referred to as “BM3D” [7]. The

enhancement of the sparsity is achieved by grouping similar 2D fragments of the image into 3D data arrays called “groups”. Due to the similarity between the grouped blocks, it enables a highly sparse representation in 3D transform domain, so that the noise can be well separated by shrinking the coefficients. However, if a template block contains geometric objects such as edges, repeating patterns, and texture, BM3D is effective only when the template block can find a large quantity of *good* matches. Due to the block searching process in BM3D, it fails to good matches when an image is heavily contaminated by noise, resulting in poor matching. Meanwhile, we would like to point out that although BM3D can generally keep fine image details while removing noise, it blurs edges after collaborative filtering. In 3D transform domain, edges can not be sparsely represented. And the inseparability between noise and edge makes the denoising filter inevitably removes partial energy of edges while mitigating noise. Furthermore, the final stage of BM3D is calculation of the weighted average of the blockwise estimates; however, weighted averaging is a low-pass filter, which blurs edges.

To mitigate the mentioned drawbacks of BM3D, we propose a bounded BM3D that can effectively remove heavy noise from images without blurring the geometric details.

1.1.2 Video Compression

With the widespread adoption of technologies such as digital television, Internet streaming video and DVD-Video, video compression has become an essential component of broadcast and entertainment media [13]. The standard of video coding for telecommunication applications has evolved through the development of the ITU-T H.261, H.262 (MPEG-2), H.263 (later enhancement known as H.263+), and H.264/AVC (MPEG4 part2) [14]. With greater needs for higher coding efficiency, VCEG is launching a long-term video coding standard H.265. Currently, H.265 hasnt been formalized and VCEG keeps seeking proposals regarding the possibility of a major gain in performance to justify H.265 from H.264. To better evaluate these contributions and retain progress, KTA (key technical area) is developed as the software platform. Among the tools

adopted in KTA, adaptive post/loop filter (ALF) is very promising in terms of both coding gain and visual quality.

As mentioned in Section 1.1.1, the functionality of filtering for video coding is two-folded: 1)improve coding efficiency; 2)remove coding artifacts. Adaptive loop filter (ALF), like in-loop deblocking filter, is designed to meet both criterions. There are three types of ALF: frame based, block based and quadtree based ALF. All these ALFs are wiener filters concatenated with deblocking filter inside the coding loop. Deblocking filter aims to remove blockiness, while ALF aims to reduce the other coding artifacts. However, if the two filters are designed independently, it may cause both to be sub-optimal. In [15], it shows that the joint design of deblocking filter and ALF is important in both performance and complexity. The challenge of filter design is how to unify deblocking filter and ALF under one framework to overcome the over-filtering problem that arises when they work sequentially but designed independently.

1.1.3 Video Hashing

The increasing availability of multimedia data in digital form has led to a tremendous growth of tools to manipulate digital multimedia. To ensure trustworthiness, multimedia authentication techniques have emerged to verify content integrity and prevent forgery [16][17]. A typical example is youtube. Huge amount of video data attempt to be uploaded to youtube every day. To prohibit the illegal distribution of copyrighted work, an automatic robust algorithm is needed to identify the content of uploaded videos, and detect the forgery of the protected work. These forgeries often go through certain manipulations such as moderate levels of filtering, geometric distortion, or noise corruption. And a robust forgery detection method for video must allow for lossy representations with graceful degradation. Therefore, unlike the original hash functions for integrity of binary text message, where hash value is highly sensitive to single bit change, we are seeking a hash more sensitive to the content rather than the exact binary representation.

Existing approaches of video hash can be classified into two categories. 1) Schemes in the first category extract features in spatial domain only. Usually, a robust image hash approach is first developed in this kind. Video hash is a simple extension of the image hash applied to those individual *key* frames extracted from the video [18][19]. And how to identify *key* frames to efficiently represent a video sequence is the most important issue of this category. Simulations show that this kind of video hash always has poor performance because key frames are only selective information of the video, and they can never truly be a good global representative of the video sequence. 2) Schemes in the second category extract features in spatio-temporal domain. These are completely independent solutions from key-frame based methods. Generally, approaches employing both spatial and temporal information tend to be more robust than key-frame based schemes because the features extracted from 3D domain are surely a better representation. Unfortunately, approaches reported in this category are very limited.

The challenge of the video hash research is to find a better way to extract 3D feature as video hash than simply applying image hash to some selected key frames, while employing the well-developed image hash methods reported in many literatures.

1.1.4 Video Communication System Delay

The compressed video stream will be delivered through network for distribution. With the development of wireless communication and dramatically increase of mobile devices use, video streaming over wireless network becomes a big concern. Examples of such applications are video conference over mobile phones and video shared on social network via smart phones and pads like iPhone®, iPad® and Android™ system based user-end. To support these applications, future wireless networks are expected to provide quality of service (QoS) guarantees, including end-to-end delay, transmission data rate, and packet error probability. Mobile devices performing video coding and transmission over wireless networks are limited in energy supply, bandwidth and time

constraint. Hence, how to assign limited resources while maximizing video quality is a valid yet difficult research topic. In Chapter 5, we put particular emphasis on end-to-end system delay performance. We briefly review the current research on delay constrained video communication system.

The challenge to meet end-to-end delay constraint is a video communication system is to understand how the time should be adaptively assigned in different modules. And to establish an accurate source coding delay-Rate-Distortion model is the very initial and essential step. For the first time, we propose a d-R-D model in source video coding to adapt encoding time, source distortion and source rate for both real time video coding as in IPPPP mode and none real time video coding as in hierarchical B pictures. In real time video coding, with d-R-D source coding model, we are able to vary transmission delay by tuning encoding parameter to change encoding time, given fixed end-to-end delay. We analyze the relationship between packet transmission error rate and encoding time through simulation, to prove the fact that the control of encoding time will influence the end-to-end system performance. For non real time video application, where B frame is employed to get better coding efficiency, we establish the d-R-D source coding model for hierarchical B pictures coding. In hierarchical B pictures, encoding time is largely dependent on the number of hierarchical B pictures level or group of picture (GOP) size. Since delay constraint is not vital in non real time coding applications, we only relate rate distortion performance of source coding with GOP size, without further analysis of end-to-end system performance.

1.2 Contributions of This Dissertation

The major contributions of our work are summarized as follows:

1. We propose a bounded block matching with 3D transform domain collaborative filtering scheme for image and video denoising.

    - We propose to partition the image to several regions before block matching. By restricting the search of similar blocks within the region of a template block, we reduce the risk of finding bad match.

- We conduct partial block matching for each coherent segment and apply shape adaptive DCT to each irregular shaped template group. It avoids applying DCT transform to edges, hence it helps keep fine image details while removing noise.
- It significantly enhance the perceptual quality of heavily noise contaminated images.

2. We propose a classified quadtree-based adaptive loop filter in video coding framework of H.264/AVC and beyond.

   - It keeps the traditional filtering framework of two in-loop filters: deblocking filter and adaptive in-loop filter working sequentially.
   - It takes the effect of deblocking filter into consideration when designing adaptive in-loop filter to overcome over-filtering problem.
   - It integrates the proposed ALF design into quadtree data structure to take advantage of previously developed QALF.
   - We investigate the choice of weighting factor with coding frame type and quantization parameter QP.

3. We propose a robust video hash scheme based on 2D-DCT temporal maximum occurrence.

   - It segments one video clip to shots, which makes it more meaningful. And it generates shot hash as video hash.
   - It extracts spatio-temporal information contained in video to generate one feature frame, and applies image hash method to this feature frame to derive shot hash. In this way, we employ well-developed image hash as what key frame method does, but achieves more robust performance.
   - We notice that for video hash applications in identification and verification, reference video and suspected video always appear in pairs. Consequently, we propose to derive shot hash in a pairwise manner. This breaks the routine to measure the quality of multimedia hash method, but achieves more flexibility.
   - It supports any play order of the video and robust to many temporal attacks.

4. We establish a source coding delay-Rate-Distortion model to study the extend Shannon's rate-distortion theory to 3 dimension with variant delay, so that to study the trade-off among source distortion, rate and encoding time.

   - Source coding d-R-D model of IPPPP mode is developed for real-time video communication applications. We analyze the system performance given

end-to-end delay when varying the encoding time in d-R-D model through experiments.

- We found the motion estimation prediction residue variance in IPPPP mode is an exponential function of motion estimation complexity parameters.
- Source coding d-R-D model of hierarchical B pictures is developed for delay insensitive video communication applications. Because bi-direction B frame is used in this coding structure, the delay is calibrated as the size of GOP in this model, rather than a single frame encoding time in IPPPP mode.
- We found the motion estimation prediction residue variance is a linear function of prediction temporal distance for both single hypothesis P frame and two hypotheses B frame.
- We proved that the source coding d-R-D model of both IPPPP mode and hierarchical B pictures is quite accurate. Although the model verification is carried out over H.264/AVC, it is not limited to any particular video compression platform.

1.3 Structure of the Dissertation

In Chapter 2, we propose a novel transform-domain based image denoising approach. It groups similar 2D fragments of one image block into 3D data arrays to enhance the sparse representation of the signal in transform domain, so that the noise can be well separated by shrinking the coefficients. This is the basic idea of block matching and 3D filtering (BM3D). However, it is well known edges cannot be sparsely represented in 3D transform domain. To preserve geometric features such as edges, repeating patterns and texture of the image, we partition the image to several regions before block matching, and the search range of similar block group is bounded within the region where the template block in located. If the template block falls in more than one region, i.e. it contains several coherent segments of arbitrary shape, grouping and shape adaptive 3D transform will apply to each coherent segment, in order to guarantee the transform will not apply across edges. The proposed bounded BM3D turns out to be particularly effective for highly noisy images. Note that video can be regarded as a collection of images, and we apply the proposed scheme to each individual frame for video denoising.

In Chapter 3, we focus on the adaptive in-loop filter (ALF) module in the emerging state-of-art encoding technology KTA/HEVC to further improve the coding efficiency. Ever sine H.264/AVC has been standardized, various filters are designed to improve the coding performance. H.264/AVC adopts deblocking filter as a in-loop filter to remove blocking artifacts and achieves both objective gain and subjective satisfaction. While KTA/HEVC proposes ALF to further enhance the filtering success. However, all existing ALFs are concatenated with deblocking filter, and the two filters work sequentially but are designed independently, which may cause both to be sub-optimal. Motivated by the joint design of the filtering framework, we propose a classified quadtree-based adaptive in-loop filter (CQALF). The output pixels of deblocking filter are classified into two categories: those modified by the deblocking filter and those untouched. A filter will be designed for each category. By taking the impact of the deblocking filter when designing the ALF, it overcomes the possible over-filtering problem.

In Chapter 4, we study the robust video hashing which applies to automatic content protection such as video identification and video authentication. We have two observations that help us formalize the idea of video hashing. 1) As a short abstract of a video content, video hashing will be meaningful only if it is derived in unit of shot. 2) For video hash applications, reference video and suspected video always appear in pairs. Hence we propose to derive shot hash in a pair-wise manner. To utilize both temporal and spatial information, 2D-DCT is applied to each frame in a shot, quantize the DCT coefficient and record the temporal occurrence of the co-located coefficient. We select a pair of closet value, inverse transform to spatial domain and derive image hashes from two feature frames by Radial hASH. Unlike key-frame based video hash and spatio-temporal domain based video hash techniques that are exclusive to each other, the proposed scheme takes advantage of both and is proven to be very robust in video identification, authentication and verification.

In Chapter 5, we turn to delay requirement in video streaming applications. A source coding model to relate encoding time, rate and distortion has been established for IPPPP coding mode and hierarchical B pictures respectively. For IPPPP mode, it is found that the motion estimation residual variance is exponentially related to motion estimation (ME) complexity parameters. On the other hand, encoding time of each P frame is function of ME complexity parameters, and R-D are functions of residual variance and thus ME complexity parameters. Therefore, d-R-D can be tuned through ME complexity parameters, and d-R-D model is developed. Next, source coding d-R-D model is integrated to end-to-end system performance, and we analyze the transmission performance given end-to-end delay by varying encoding time in d-R-D model. For hierarchical B pictures, where end-to-end delay is much less demanding than in real-time applications, encoding time d is expressed in group of picture (GOP) size, which can be mapped to prediction temporal distance for each individual hierarchical level. Meanwhile, we discover that the prediction residual variance is linear function of prediction temporal distance, and thus rate and distortion can be written as functions of prediction temporal distance. Experimental results show that our proposed d-R-D model for both IPPPP mode and hierarchical B pictures are accurate enough when the method of compensation from neighboring frame/GOP is applied.

Finally, Chapter 6 concludes the dissertation. We also propose some research topics which might be our potential future work.

CHAPTER 2
IMAGE DENOISING BY BOUNDED BLOCK MATCHING AND 3D FILTERING

In this chapter, we propose a bounded BM3D scheme. The novelty of our bounded BM3D is two-folded. First, our scheme partitions an image into multiple regions, and identifies the boundaries between regions. And we restrict block matching search within the region of the template block. Second, to prevent important geometric features such as edges from being removed by collaborative filtering in BM3D, we do partial block matching for different block coherent segments which belong to different regions. The bounded BM3D further improves the denoising performance for heavily contaminated noisy image and video than the original BM3D.

2.1 Research Background

The search for efficient image denoising methods still poses a challenge. All the existing methods rely on some explicit or implicit assumptions about the true (noise-free) signal in order to separate it from the noise [1]. In particular, the transform-domain denoising typically assumes that the signal is sparsely represented in the transform domain. However, the great variety of natural images makes it impossible for any fixed 2D transform to achieve good sparsity for all cases. Recently, a novel image denoising strategy based on an enhanced sparse representation in transform domain is introduced in[20][7], referred to as “BM3D”. The enhancement of the sparsity is achieved by grouping similar 2D fragments of the image into 3D data arrays called “groups”. Due to the similarity between the grouped blocks, it enables a highly sparse representation in 3D transform domain, so that the noise can be well separated by shrinking the coefficients. To the best of our knowledge, BM3D achieves the best performance for removing additive white Gaussian noise at a reasonable computational cost, in terms of both mean-squared-error (PSNR) and subjective visual quality [21][22].

However, if a template block contains geometric objects such as edges, repeating patterns, and texture, BM3D is effective only when the template block can find a large

quantity of *good* matches. We notice that BM3D allows block-matching to search out of the region that contains the template block. For a less noisy image, a wide search range may result in crossing different regions that contain different patterns; this increases the chance of finding more good matches. However, when an image is heavily contaminated by noise, true image features are usually concealed in noise, resulting in poor matching. To mitigate this drawback, we propose to partition the image to several regions before block matching. And by restricting the search within the region of a template block, we reduce the risk of finding bad match.

Meanwhile, we would like to point out that although BM3D can generally keep fine image details while removing noise, it blurs edges after collaborative filtering. In 3D transform domain, edges can not be sparsely represented. And the inseparability between noise and edge makes the denoising filter inevitably removes partial energy of edges while mitigating noise. Furthermore, the final stage of BM3D is calculation of the weighted average of the blockwise estimates; however, weighted averaging is a low-pass filter, which blurs edges. In contrast, in our bounded BM3D, if a block contains edge and belongs to more than one region, we conduct partial block matching for each coherent segment rather than the block as a whole. (Here a coherent segment refers to the irregular shaped portion of a block that matches some portion in a template block; e.g., S1 in Figure 2-3 is a coherent segment matched to some portion in a template block; S2 in Figure 2-3 is another coherent segment matched to some portion in a different template block.) After partial block matching, we apply shape adaptive DCT [23][24] to each irregular shaped template group. In this way, our method preserves edges and avoids applying DCT transform to edges since DCT is ineffective in representing edges [25].

2.2 The Bounded BM3D

Similar to BM3D, bounded BM3D follows a two-stage blockwise scheme. In the first stage, we group the matching blocks for every template block in a *noisy* image, and

apply hard-thresholding filtering to get a basic estimate. In the second stage, we group the matching blocks for every template block in the *basic* estimate, and apply Wiener filtering to produce the final estimate. Figure 2-1 illustrates the two stage schemes in a graphic diagram. The difference between BM3D and bounded BM3D is two-folded. 1) After the first stage, the basic estimate is partitioned to regions, and the image boundaries are detected. 2) In the second stage, based on the detected boundaries, we conduct partial block matching for each coherent segment, and restrict search range of block matching within the region of the template block. Then we apply shape adaptive DCT to each coherent segment in Wiener filtering. If a coherent segment happens to be a square block, the procedure is the same except shape adaptive DCT falls back to block DCT. The reason we partition the basic estimate other than the input noisy image is the lack of accuracy in detecting edges in noisy images. Hence partial block matching can only be implemented in Wiener filtering stage where a denoised "cleaner" image - basic estimate is available. We process all overlapping blocks of fixed size in a sliding manner, where process refers to block-matching and filtering in 3D transform domain.

We consider a noisy image $z : \mathbf{X} \to \mathbb{R}$ of the form

$$z(\mathbf{x}) = y(\mathbf{x}) + \eta(\mathbf{x}), \mathbf{x} \in \mathbf{X} \tag{2–1}$$

where $\mathbf{x}$ is a 2D spatial coordinate that belongs to the image domain $\mathbf{X} \subset \mathbb{Z}^2$, y is the true image, and η is i.i.d. zero-mean Gaussian noise with variance σ^2.

With $Z_{\mathbf{x}}$ we denote a fixed size block $N_1 \times N_1$ extracted from z, where $\mathbf{x}$ is the coordinate of the top-left corner of the block. Alternatively, we say that $Z_{\mathbf{x}}$ is located at $\mathbf{x}$ in z. A group of collected 2D blocks is denoted by a bold-face capital letter with a subscript that is the set of its grouped blocks' coordinates, e.g. $\mathbf{Z}_s$ is a 3D array composed of blocks $Z_{\mathbf{x}}$ located at $\mathbf{x} \in S \subseteq \mathbf{X}$. In order to distinguish between parameters used in the first and in the second stage, we respectively use the superscripts "ht"

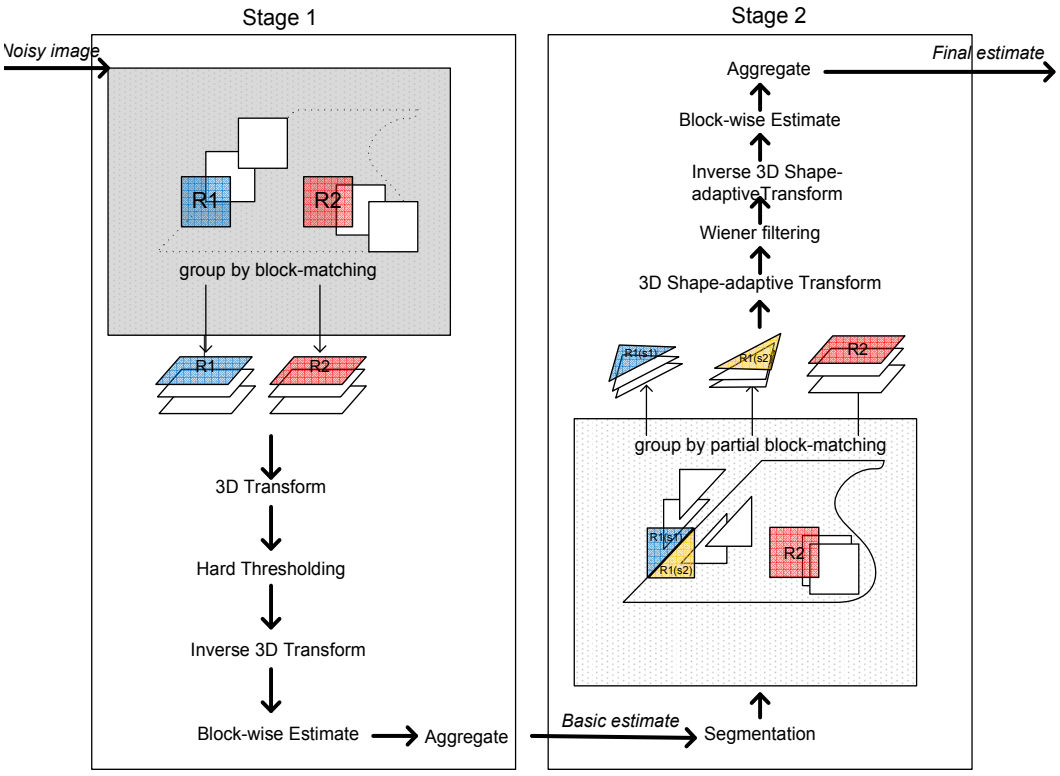


Figure 2-1. Diagram of bounded BM3D. Dashed lines in noisy image indicates image boundary is hidden in noise, and it is detected after segmentation in basic estimate.

(hard-thresholding) and “wie” (Wiener filtering). We will present the steps of the proposed bounded BM3D in detail in the following parts.

2.2.1 Hard Thresholding Estimate

We employ hard-thresholding filtering to process the input noisy image to get the basic estimate. This stage follows exactly the same procedures with those in BM3D.

For a reference block located at $\mathbf{x}_R$, $Z_{\mathbf{x}_R}$, its matching blocks group to a stack,

$$S_{x_R}^{ht} = \{\mathbf{x} \in \mathbf{X} : d(Z_{x_R}, Z_x) \leq \tau^{ht}\} \tag{2–2}$$

$d(\mathbf{Z}_{\mathbf{x}_R}, \mathbf{Z}_{\mathbf{x}})$ measures the block distance between the template block and the candidate search block in noisy image, which employs a hard-thresholding prefiltering [7], and τ^{ht} is the distance threshold within which two blocks match. On the other hand, if no match is found, which implies $S_{\mathbf{x}_R}^{ht}$ is null, the method falls back to a 2D problem.

In general, the block-wise estimates can overlap [7], and each block may have more than one estimate. Hence, the basic estimate for $Z_{\mathbf{x}_R}$, denoted as $\hat{Y}^{ht}_{\mathbf{x}_R}$, is a weighted aggregation of all the block estimates located at $\mathbf{x}_R$.

2.2.2 Image Boundary Detection

Once we get the basic estimate, we can detect the boundaries by partitioning the basic estimate. The partition method used is left open in our scheme. Experiments show that the selection of partition approach does not have a great impact on the performance gain demonstrated in bounded BM3D. Consequently, in this paper, we employ computational efficient k-means clustering method to partition the basic estimate and get the boundaries thereafter. Figure 2-2 shows the k-means clustering results of 512x512 images Lena and Barbara, where $k = 4$, and each cluster is indicated by a certain gray level that is easy to distinguish in the images.

2.2.3 Wiener Filtering with Image Boundary

Provided that an estimate of the true image is available (e.g. the basic estimate from the hard-thresholding), we can construct an empirical Wiener filter as a natural extension of the thresholding technique [7]. Since the major steps of Wiener filtering in bounded BM3D follows the same procedure as in BM3D, we only give the fundamental modifications required and omit the repetition of the concept. Let us denote the basic estimate by $\hat{y}^{ht} : \mathbf{X} \rightarrow \mathbb{R}$. $E_{\mathbf{x}}$ designates a square block of fixed size $N_1 \times N_1$ extracted from $\hat{y}^{ht}$ and located at $\mathbf{x} \in \mathbf{X}$.

2.2.3.1 Block-matching

In this stage, block matching is conducted in the basic estimate $\hat{y}^{ht}$ rather than the noisy image. Since we restrict the block matching search within the same range of template block, we should first check if the template block contains more than one coherent segment. All the blocks fall in two categories: block contains only one segment (coherent segment is the whole square block) and block contains more than one coherent segment. In the first category, we conduct the whole block matching within the

Figure 2-2. Image clustered by k-means to 4 clusters, indicated by different gray level. (a) 512x512 Lena; (b) Clustered Lena; (c) 512x512 Barbara; (d) Clustered Barbara.

region of the reference block, and cross boundary search is forbidden. In the second category, partial block matching is done for each coherent segment with the help of a mask, which is of the exact same shape as the coherent segment, as illustrated in Figure 2-3. By doing partial block matching for individual coherent segment that does not contain boundaries, we avoid dealing with edges, so that we also avoid the sparse representation problem of edges in 3D transform domain later in Wiener Filtering.

Suppose a block contains two coherent segments, denoted as S_1, S_2, and $m(S_1)$, $m(S_2)$ are masks corresponding to S_1 and S_2 respectively. By multiplying the candidate block with the mask, every match block will be of the same shape as the coherent

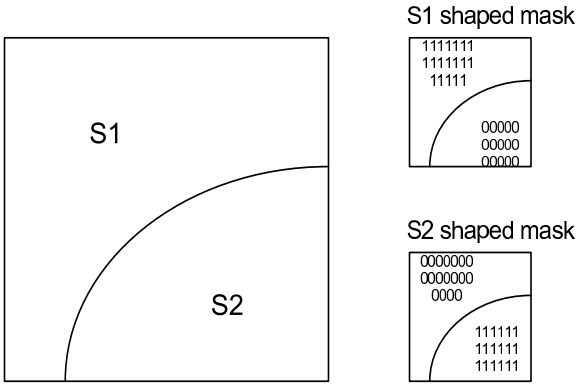


Figure 2-3. Mask formation for a block containing two coherent segments S_1 and S_2.

segment, and a group of S_1, S_2 shaped blocks are collected for 3D transform denoising later. The block distance measure used in block matching for S_i is written as

$$d_{S_i}(E_{\mathrm{x}_R}, E_{\mathrm{x}}) = \frac{\|E_{\mathrm{x}_R} \bigcap m(S_i) - E_{\mathrm{x}} \bigcap m(S_i)\|^2}{\#m(S_i)^2} \tag{2–3}$$

where E_{x_R} is the template block located at $\mathbf{x}_R$, E_{x} denotes the candidate matching block located at $\mathbf{x}$, and $\#m(S_i)$ is the number of non-zero elements in $m(S_i)$. Hence the union of $m(S_i)$ with candidate block screens out the elements that are out of the shape of the coherent segment in that candidate block.

Hence, the matched blocks for coherent segment S_i are

$$S^{wie}_{\mathrm{x}_R,i} = \{\mathbf{x} \in \mathbf{X} : d_{S_i}(E_{\mathrm{x}_R}, E_{\mathrm{x}}) < \tau^{wie}\} \tag{2–4}$$

For expression simplicity, we use s_i to replace $S^{wie}_{\mathrm{x}_R,i}$ and leave out the superscript "wie" in the symbols below, where it does not cause any confusion.

With matching block set for S_i, we get two matching groups from basic estimate and noisy image for S_i:

1) $\hat{\mathbf{Y}}^{ht}_{s_i}$ by stacking together the basic estimate fragments $\hat{Y}^{ht}_{s_i}$;

2) $\mathbf{Z}_{s_i}$ by stacking together the noisy image fragments Z_{s_i}.

2.2.3.2 Wiener filtering

We define the empirical Wiener shrinkage coefficients from the 3D transform coefficients of the initial estimate group as:

$$\mathbf{W}_{s_i} = \frac{|\Gamma_{3D}(\hat{\mathbf{Y}}^{ht}_{s_i})|^2}{|\Gamma_{3D}(\hat{\mathbf{Y}}^{ht}_{s_i})|^2 + \sigma^2}. \tag{2–5}$$

The collaborative Wiener filtering is applied to the noisy image fragment $\mathbf{Z}_{s_i}$. It is done in the manner of element-wise multiplication of the 3D transform coefficients $\Gamma_{3D}(\mathbf{Z}_{s_i})$ of the noisy data with the Wiener shrinkage coefficients $\mathbf{W}_{s_i}$. Subsequently, the inverse transform $(\Gamma_{3D})^{-1}$ produces the group of estimates

$$\hat{\mathbf{Y}}^{wie}_{s_i} = (\Gamma_{3D})^{-1}(\mathbf{W}_{s_i}\Gamma_{3D}(\mathbf{Z}_{s_i})) \tag{2–6}$$

located at the matched locations $\mathbf{x} \in s_i$.

To aggregate the overcomplete estimates, for each $\mathbf{x}_R \in \mathbf{X}$, we assign the weight

$$w_{\mathrm{x}_R,i} = \sigma^{-2}\|\mathbf{W}_{s_i}\|^{-2} \tag{2–7}$$

for the group of estimates $\hat{\mathbf{Y}}^{wie}_{s_i}$.

Let $\hat{Y}^{\mathrm{x}_R}_{\mathrm{x}_m}(x)$ be an estimate of $y^{wie}(x)$, where $\mathbf{x}_m \in S^{wie}_{\mathrm{x}_R,i}$. We extend the arbitrary shaped support of $\hat{Y}^{\mathrm{x}_R}_{\mathrm{x}_m}(x)$ by zero to simplify the formulation. And the final estimate of the noisy image y^{wie} is the weighted average of all estimates located at the same place, and computed by

$$\hat{y}^{wie}(x) = \frac{\sum_{\mathrm{x}_R}\sum_{\mathrm{x}_m} w_{\mathrm{x}_R,i}\hat{Y}^{\mathrm{x}_R}_{\mathrm{x}_m(\mathrm{x})}}{\sum_{\mathrm{x}_R}\sum_{\mathrm{x}_m} w_{\mathrm{x}_R,i}\chi_{\mathrm{x}_m}(\mathbf{x})} \tag{2–8}$$

where $\chi_{\mathbf{x}_m} : \mathbf{X} \to \{0, 1\}$ is the characteristic function of the square support of a block located at $\mathbf{x}_m \in \mathbf{X}$.

The problem arises when conducting 3D transform on mask-shaped block group in (2–5), where the rectangular shaped block transform is often not applicable.

The choice of the 3D transform Γ_{3D} is governed by their energy compaction (sparsity) ability for noise-free image block groups. Experiments in [7] show that 3D transform can be separated to 2D+1D transform, where the well-established decorrelating transforms, such as DCT, DFT, and wavelets can be used for 2D transform, and Haar for 1D transform. In this paper, we choose DCT as 2D transform, and introduce computational attractive shape-adaptive DCT (SA-DCT) to deal with the arbitrary shaped match blocks for each coherent segment.

It is computed by cascaded application of 1D varying-length DCT transforms first on the columns and then on the rows that constitute the considered region [24]. In terms of the basis elements, they are defined as:

$$\Psi_{L,m}^{1D-DCT}(n) = c_m cos(\frac{\pi(2n+1)}{2L}), m, n = 0, \cdots L-1. \tag{2–9}$$

$$c_0 = \sqrt{1/L}, c_m = \sqrt{2/L}, m > 0 \tag{2–10}$$

Here L stands for the length of the column or row to be transformed. And SA-DCT will degenerate to rectangular 2D-DCT transform if the support is rectangular shaped.

Note that the template block that contains only one coherent segment is a special case of the computation above, with $S_i, i = 1$ only. The mask is simplified to square shape, and hence rectangular shaped block matching and block 2D-DCT is conducted.

2.3 Experimental Results

We have stated in Section 2.2 the limitation of BM3D in highly noise-contaminated image and why bounded BM3D can improve the denoising performance in this case. Therefore, the experiments in this section focus on heavily noisy images. Table 2-1 compares the PSNR (dB) performance of BM3D (software available in [26] with bounded BM3D in video frames *foreman* and *trevor*, with noise level at $\sigma = 50$ and $\sigma = 100$, respectively. The proposed bounded BM3D achieves 0.23dB-1.18dB gain in these two cases. In Figure 2-4, we present the error-free image, heavily noisy image ($\sigma = 100$) and

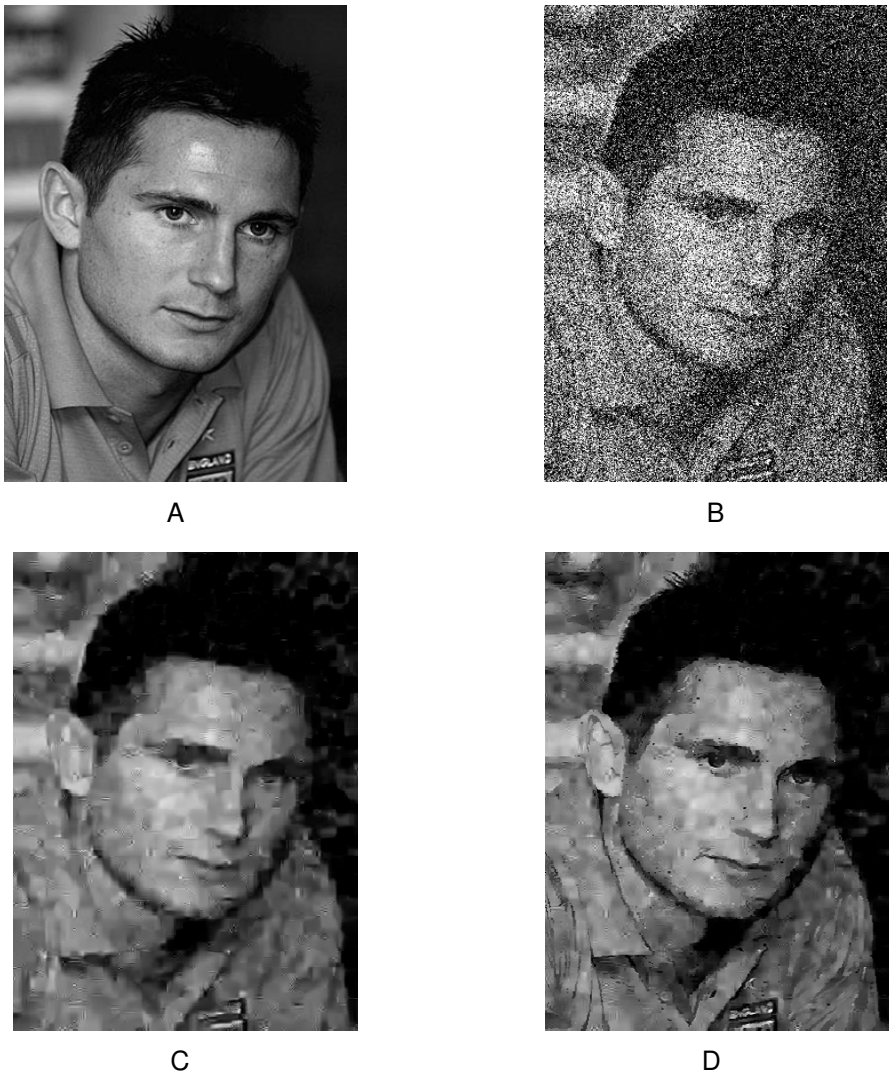


Figure 2-4. Denoising performance comparison between BM3D and Bounded BM3D with heavy noise. (a) Noise-free image. (b) Noisy image with $\sigma = 100$. (c) Denoised image using BM3D,PSNR=23.54dB. (d) Denoised image using bounded BM3D, PSNR=24.06dB.

the denoise results using BM3D and bounded BM3D respectively. Similarly, bounded BM3D improves the PSNR by 0.52dB. More remarkably, it shows much improvement in the visual performance, where geometric objects such as edges and texture are more visible to human eyes. We also apply the two schemes in standard test images,*Barbara*(512x512),*Lena* (512x512) and *House* (256x256), to further validate the improvement of the proposed bounded BM3D over BM3D. Figure 2-5 gives the enlarged fragments of

Table 2-1. PSNR(dB) comparison between bounded BM3D and BM3D for video frames with heavy noise

	$\sigma = 50$			$\sigma = 100$		
	PSNR (dB)		PSNR increased by	PSNR (dB)		PSNR increased by
	BM3D	Bounded BM3D	Bounded BM3D	BM3D	Bounded BM3D	Bounded BM3D
foreman	25.68	26.23	0.55	21.85	23.03	1.18
trevor	25.30	25.53	0.23	21.66	22.64	0.98

Table 2-2. Running Time comparison between bounded BM3D and BM3D for video frames with heavy noise

	Running Time (sec)		Time increased by
	BM3D	Bounded BM3D	Bounded BM3D
foreman	0.40	0.48	18.96%
trevor	0.42	0.47	11.77%

the denoised images, as well as the PSNR with the two schemes. We see that bounded BM3D presents better visual effect in texture (the line pattern in Barbara) and object boundary (the boundary in House), as we have expected it achieve. It improves PSNR value in three test images of 0.92dB, 1.30dB and 1.33dB respectively.

To better evaluate the computational complexity of bounded BM3D, we compare the running time (in second) of BM3D and bounded BM3D of the experiment above ($\sigma = 50$). We run the Matlab simulation of the two algorithms on a platform of an Intel Core2 Duo CPU (@2.4GHz and 2.39GHz) PC with MS Windows Vista and 4 GB memory. The results are shown in Table 2-2. It can bee seen that bounded BM3D has a time increase of 15.37% over BM3D on average. The complexity increase is mainly due to the additional step of image boundary detection.

2.4 Bounded BM3D Extension

Up to now, bounded BM3D has only been applied to grayscale images. We would like to extend the method to color image denoising. Given a RGB color image, the most straightforward way is to apply bounded BM3D to each single channel independently. More generally, the RGB image will be firstly transformed to a luminance-chrominance color space, such as YCbCr, opponent, YIQ etc, and apply denoising separately in

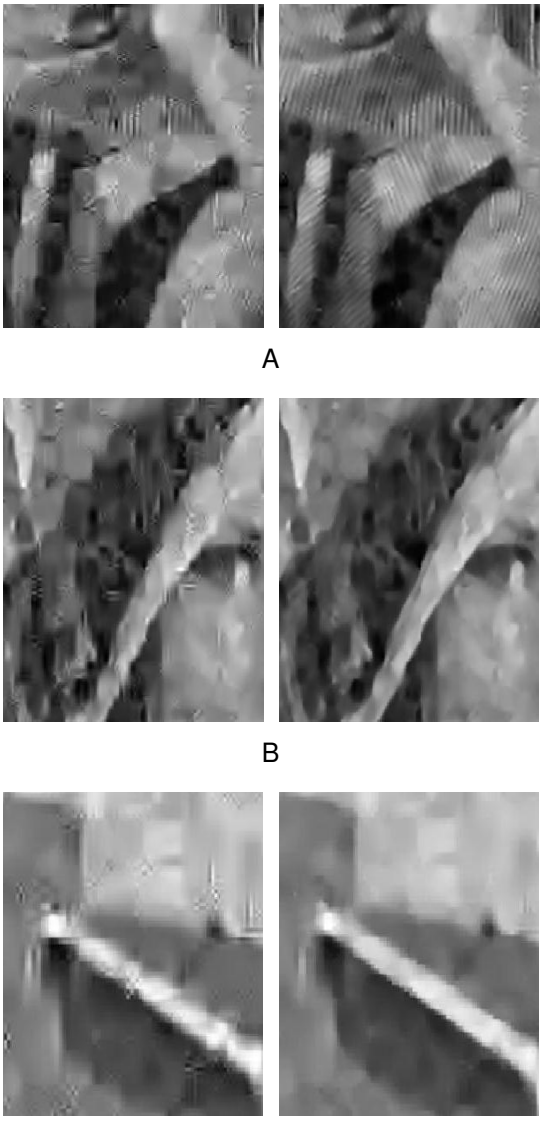


Figure 2-5. Fragments of denoised images with BM3D and Bounded BM3D in heavy noise $\sigma = 100$. The left result is by BM3D, the right is by bounded BM3D. (a)Barbara, PSNR=21.44dB vs.PSNR=22.36dB. (b) Lena, PSNR=23.76dB vs.PSNR=25.06dB. (c) House,PSNR=23.53dB vs.PSNR=24.86dB.

each channel. However, the results in [27] reported that this approach suffers from the lower SNR in the chrominances since the grouping is quite sensitive to the level of noise. Instead, it imposes a grouping constraint on the chrominances, so that grouping is performed only once in luminance channel, and is reused exactly for chrominance channels. This is based on the assumption that if the luminances of two blocks are similar, then their chrominances are also similar, which is true for most natural images. Actually, the idea of filtering the chrominance using information from the luminance was exploited already in the Pointwise SA-DCT denoising method in [28][29]. We can borrow the same idea here to group blocks in luminance, and reuse the grouping information to chrominances. All the other procedures follow exactly the bounded BM3D algorithm in section 2.2. Furthermore, given that grouping by block-matching takes approximately half of the execution time of the BM3D, In addition, considering grouping by block-matching takes approximately half of the execution time in bounded BM3D, the grouping constraint enables a computational reduction of approximately one third as compared to applying the bounded BM3D separately on each channel [27].

Ideally, as a improvement of original BM3D, bounded BM3D can replace BM3D and be extended to all applications where BM3D are employed. These include but not limit to: 1)video denoising in 3-D nonlocal spatio-temporal transform [30], notice the video denoising is not conducted frame by frame as image denoising, but rather group a 3-D cube from spatio-temporal search; 2)joint image and sharpening [31][32]; 3)image deblurring [33][34][35]; 4)image and video up-sampling and super-resolution [36][37]; 5) joint denoising of multiframe raw data from a digital imaging sensor[38]; 6)color image reconstruction from noisy color filter array [39]; 7)video denoising in 4-D nonlocal spatio-temporal transform [40].

2.5 Summary

In this chapter, we proposed a bounded block-matching with 3D transform domain collaborative filtering method to effectively mitigate noise from images that are highly

noisy. The novelty of our bounded BM3D is two-folded. First, after the hard-thresholding filtering, our scheme partitions the basic estimate into multiple regions, and identifies the boundaries between regions. And we restrict block matching search within the region of the template block. Second, to prevent important geometric features such as edges from being removed by collaborative filtering in BM3D, we do partial block matching for different coherent segments which belong to different regions in Wiener Filtering extension. Compared with BM3D, the proposed bounded BM3D achieves better subjective performance, as well as 0.23dB-1.33dB PSNR increase for heavily noisy images.

It should be noted that in the experiments, the white Gaussian noise is added to corrupt images manually. Hence we know the noise variance in advance. However, in real denoising applications, where the noise profile of the noisy image is unknown, certain estimation strategy is needed [41][42][43][44][45][46] to first estimate the noise variance before we can apply bounded BM3D.

CHAPTER 3
CLASSIFIED QUADTREE-BASED ADAPTIVE LOOP FILTER

3.1 Research Background

Many techniques have been developed to improve the coding performance of H.264/AVC for high definition video contents. Among these techniques, the adaptive in-loop filter (ALF) is proposed and achieves an attractive gain over H.264/AVC. The basic idea of ALF is to estimate one or multiple filters at encoder by restoring the reconstructed frame towards its original version. This technique can effectively remove more general coding artifacts when working together with deblocking filter and prepare better reference frames for coding future frames, therefore improving the coding performance.

In [47], the wiener filter is used as the post filter to improve the subjective and objective qualities. [48] and [49] are two earliest proposals of ALF schemes to place the wiener filter inside the coding loop. Although the filter is temporally adaptive by updating the filter coefficients frame by frame, it is usually kept spatially invariant. To achieve better adaptation, several techniques are proposed in recent academic and standardization activities.

In [50][51][52], a block-based adaptive in-loop filter (BALF) was proposed. The reconstructed frame is partitioned to non-overlapped and equal-size blocks. For each block, a flag is used to signal whether the block will be filtered or not. Quadtree-based adaptive in-loop filter (QALF) further improves the filter performance by allowing blocks to be of variable size, and provides a signaling scheme in a quadtree data structure [53]. Based on QALF, multiple filters scheme is proposed by estimating a set of filters [54] instead of only one filter in QALF. Whenever the QALF partition is indicated to be filtered, for each pixel in the block, a specific filter from the set is chosen based on a measure of local characteristic of the pixel. The filters take local characteristic of a frame into account and allow more spatial adaptation. In [55], the concept of multiple inputs

filter was proposed by taking the reconstructed frame, prediction frame, and prediction residual frame as the inputs of the filter in order to incorporate more hypothesis and consider different characteristics of input signals for restoration. It automatically adapts the filter coefficients to different inputs during the filter estimation.

Although the wiener filter can efficiently remove some artifacts that are caused during compression, it is less efficient to remove blocky artifacts compared with the traditional deblocking filter. So in all previous works, the wiener filter is designed after the deblocking filter. These two filters work sequentially but are designed independently, which may cause both to be sub-optimal. In [15], it shows that the joint design of deblocking filter and ALF is important in both performance and complexity.

Motivated by the joint design of the filtering framework, in this paper, we propose a classified quadtree-based adaptive in-loop filter (CQALF) that takes advantage of both multiple filters and multiple inputs filter schemes. The proposed scheme is based on the hybrid video coding framework, for example H.264/AVC. Instead of unifying the deblocking filter and adaptive in-loop filter into one filter as in [15], we keep the traditional structure which makes the problem simpler. However, the impact of the deblocking filter of H.264/AVC is considered during the wiener filter restoration. The output pixels of deblocking filter are classified into two categories: those modified by the deblocking filter and those untouched. A filter will be designed for each category. For the pixels modified by the deblocking filter, the adaptive loop filter takes two inputs, frames before and after the deblocking filter. A weighted average of the two inputs are used to estimate the filter coefficients, which avoids increasing the number of filter coefficients and hence saves the coding overhead. For the pixels which are not modified by deblocking filter, only the frame after the deblocking filter is exploited to estimate the filter coefficients. CQALF classifies pixels by taking the impact of the deblocking filter into account in order to avoid the possible over-filtering problem. No overhead is needed

to indicate the classification of each pixel because the classification is based on the reconstructed frame which is available at both encoder and decoder.

3.2 Deblocking Filter and Adaptive in-loop Filter

3.2.1 Deblocking Filter

In H.264/AVC, a deblocking filter is exploited to remove the blocky artifacts around the block boundaries in order to improve the quality of a coding frame [56][57][58] and achieve improved subjective and perceptual quality [59]. To preserve image details and remove blockiness, a boundary strength (Bs) parameter is assigned to every boundary between two blocks. If filtering conditions are met, a pre-defined filter will be applied. Filtering does not take place for edges with $Bs = 0$. For nonzero Bs values, filtering on the edge only takes place when the following 3 conditions are met:

$$|p_0 - q_0| < \alpha(Index_A) \tag{3–1}$$

$$|p_1 - p_0| < \beta(Index_B) \tag{3–2}$$

$$|q_1 - q_0| < \beta(Index_B) \tag{3–3}$$

where p_0, q_0 are block boundary pixels located in Figure 3-1, α and β are dependent on the quantization parameter (QP) employed over the edge as well as the encoder offset value to control the properties of the deblocking filter on slice level [57]. The deblocking filter can successfully remove blocky artifacts. It is not efficient to remove more general artifacts, like ringing artifacts, blurring, and so on, which may be produced during the compression process.

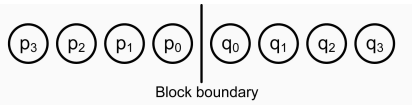


Figure 3-1. Notation of pixels across a block boundary.

3.2.2 Adaptive in-loop Filter

In recent video coding standardization activities, a wiener filter based adaptive in-loop filter (ALF) was proposed in order to further improve the quality of the reconstructed frame. The basic idea of ALF is to estimate one or multiple filters at encoder by restoring the recontracted frame towards its ordinal version. This technique can effectively remove more general coding artifacts when working together with deblocking filter and prepare better reference frames for coding future frames, therefore improving the coding performance.

In [47], the wiener filter was used as the post filter to improve the subjective and objective qualities. [48] and [49] are two earliest proposals of ALF schemes to place the wiener filter in the coding loop. Although the filter is temporally adaptive by updating the filter coefficients frame by frame, it usually keeps spatially invariant. To achieve better adaptation, several techniques are proposed in recent academic and standardization activities.

In [50] and [52], a block-based adaptive in-loop filter (BALF) was proposed, with its improvement in [60]. The recontracted frame is partitioned by non-overlapped and equal-size blocks. For each block, a flag is used to signal whether the block will be filtered or not. Quadtree-based adaptive in-loop filter (QALF) further improves the filter performance by allowing blocks to be different size, and provides a signaling scheme in a quadtree data structure[61][53], with its improvement in [62]. Based on QALF, multiple filter scheme is proposed by estimating a set of filters [54][63] instead of only one filter in QALF. Whenever the QALF partition is indicated to be filtered, for each pixel in the block, a specific filter from the set is chosen based on a measure of local characteristic of the pixel. The filters take local characteristic of a frame into account and allow more spatial adaptation. In [55], the concept of multiple inputs filter was proposed by taking the reconstructed frame, prediction frame, and prediction residual frame as the inputs of the filter in order to incorporate more hypothesis and consider different characteristics

of input signals for restoration. It automatically adapts the filter coefficients to different inputs during the filter estimation.

Suppose s is a pixel in the original frame; x_0 is its corresponding degraded pixel in the reconstructed frame; $X_0 = [x_0, x_1, \cdots, x_{n-1}]^T$ is a pixel set that consists of the pixels surrounding x_0 with a defined filter support $\mathcal{N}$. The filtered pixel of x_0 can be formulated as

$$\hat{x}_0 = \sum_{i=0}^{n-1} w_i x_i = H^T X_0 \tag{3–4}$$

where $H = [w_0, w_1, \cdots, w_{n-1}]^T$ is the filter coefficients. Assume the filter is spatial invariant, the wiener filter is the one which minimize the mean square error of the whole frame:

$$MSE = \sum_{i=0}^{M} \|s_i - \hat{x}_i\|_2 = \sum_{i=0}^{M} \left\|s_i - H^T X_i\right\|_2 \tag{3–5}$$

where M is the number of pixels in the frame. The wiener filter can be estimated by:

$$H^* = \underset{H}{arg\,min} MSE = (C^T C)^{-1} C^T S \tag{3–6}$$

where $C = [X_0, X_1, \cdots, X_M]^T$, $S = [s_0, s_1, \cdots, s_M]^T$.

The filter coefficients will be updated frame by frame to achieve the temporal adaptation. A quadtree based filtering signaling scheme is proposed to turn on/off the filtering operation at different regions.

3.3 Classified Quadtree-based Adaptive in-loop Filter

3.3.1 Problem Formulation

In KTA or recent released TMuC software, the deblocking filter and adaptive in-loop filter is concatenated to filter the reconstructed frame sequentially but the two filters are designed independently. Under this strategy, some pixels may be filtered twice, which may cause the over-filtering. The over-filtering can degrade the performance of both filters, thus these filters should design and work jointly in order to achieve the best possible performance.

In this paper, we propose to consider the deblocking filter and ALF jointly by classifying the pixels into two categories: pixels that are modified by deblocking filter and pixels that are untouched by deblocking filter. To avoid the over-filtering, the idea is to design a different filtering operation for the pixels that have already been filtered by the deblocking filter. Let us denote the original input frame as s, the reconstructed frame before deblocking filter as s', and the one after deblocking filter as s'' in the following of the paper. According to the filtering rule of deblocking filter, only some pixels (especially around the block boundaries) are filtered, while the rest are unchanged. The pixel classification is done by comparing the frame s' and s'' pixel by pixel. We denote Category I as the pixels which are modified by the deblocking filter and Category II as the pixels which are not modified by the deblocking filter.

The coding diagram of CQALF is illustrated in Figure 3-2. For Category I, each pixel has two hypothesis from s' and s''. We take both of them as the input of the filter corresponding to Category I. To avoid too much additional overhead, we apply the same set of filter coefficient to both inputs instead of using two sets of filter coefficients as in [55]. To handle the two inputs differently based on their characteristics, we propose to combine them by weighting average, which is controlled by a weighting factor ω. Based on off-line training, we found that there is a relationship between weighting factor and quantization parameter (QP) for each frame type, which will be elaborated in Section 3.3.2. For the pixels in Category II, only one input s'' is used. Though either s' or s'' can be used for input as they are exactly equal for every pixel to be filtered, we choose to use s''. This is because the surrounding pixels within its filter support $\mathcal{N}$ are generally of better quality in s'' than in s', which is based on the observation that the reconstructed frame after deblocking filter shows overall improvement in both objective and subjective quality than that before deblocking filter.

Suppose filter length is n for both filters, the filter of Category I is $H_1 = [w_{10}, w_{11}, \cdots, w_{1,n-1}]^T$, the filter of Category II is $H_2 = [w_{20}, w_{21}, \cdots, w_{2,n-1}]^T$, input

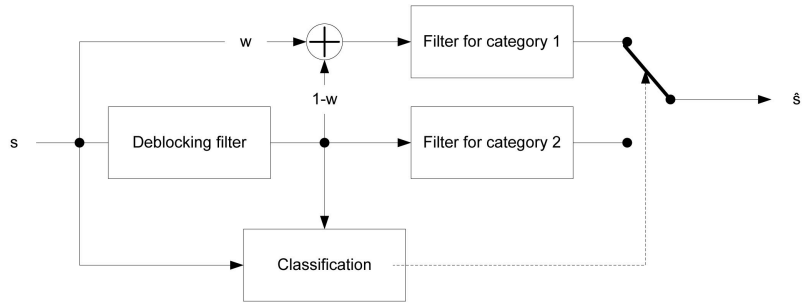


Figure 3-2. Block Diagram of Encoder with CQALF.

from s' as $X = [x_0, x_1, \cdots, x_{n-1}]^T$, input from s'' as $Y = [y_0, y_1, \cdots, y_{n-1}]^T$, and $\hat{s}$ as the output of the filter. For Category I, a pixel will be filtered as

$$\hat{s}_i = H_1^T(\omega X_i + (1-\omega)Y_i) \tag{3–7}$$

the filter can be estimated by

$$H_1 = \underset{H_1}{arg\,min} \sum_{i=0}^{M_1} \left\| s_i - H_1^T(\omega X_i + (1-\omega)Y_i) \right\|_2 \tag{3–8}$$

where M_1 denotes the number of pixels in Category I. The pixels in Category II will be filtered as

$$\hat{s}_i = H_2^T Y_i \tag{3–9}$$

the filter can be estimated by

$$H_2 = \underset{H_2}{arg\,min} \sum_{i=0}^{M_2} \left\| s_i - {H_2}^T Y_i \right\|_2 \tag{3–10}$$

where M_2 denotes the number of pixels in Category II.

3.3.2 Parameter Settings

It can be seen in (3–1) that the deblocking filtering strength is controlled by boundary strength and two parameters, α and β. The filtering process only takes place when the difference between neighboring pixels is bigger than corresponding threshold α or β. Their values increase with QP and are obtained from a look-up table which is specified by H.264/AVC. In particular, at the low end of table where $Index_A < 16$

and $Index_B < 16$, one or both α and β are clipped to 0 and filtering is turned off [57]. That implies few pixels will be modified in deblocking filter due to small α and β when QP is very small. It results in few different pixels in s' and s''. Under such circumstances, Category I is almost empty which leads to CQALF being the same algorithm as ALF.

To improve the overall performance, we observed that to increase the deblocking parameter α and β slightly will give better result in relatively low QP range. Similar strategy has also been reported in [64]. In our algorithm, the better case of the two: with or without increasing deblocking parameter, is chosen at encoder based on the rate-distortion (RD) cost. A flag at slice header is used to signal the selected deblocking parameter setting if CQALF is enabled. Simulation results show that to choose α and β on an RD basis will have BD bitrate saving of 0.8% for IPPP mode and 0.58% for hierarchical-B (HB) mode.

As proposed in CQALF, the filter corresponding to Category I has two inputs. A weighting factor ω is used to combine them as in Eq.(3–7). When ω varies from 0 to 1, input s' will gradually get involved in the filter. Considering s' is usually of lower quality than s'', how much it is involved should influence the performance of CQALF. The off-line training shows that for P and B frames, the coding gain reaches its peak at $\omega = 0.5$ regardless of QP.

On the other hand, he quality of s' differs significantly in intra frame with QP. Hence we should analyze the effect of QP when determining the best ω for intra frame. We test the intra only mode for all test sequences described in Section 4.4 of 50 frames at QP from 22 to 37 with a step of 5. For each QP point, we vary ω from $0 - 1$, select the ω that give the highest gain, and plot QP vs. ω. Figure 3-3 shows the average result over all test sequences. It can be seen that the optimal ω drops with QP's increase. When QP is small, the quality of s' does not differ much from s'', i.e. the quality improvement of deblocking filter is not significant. Meanwhile, small QP is more likely to have slight increase of α, β, so that more pixels will be modified by deblocking filter and makes

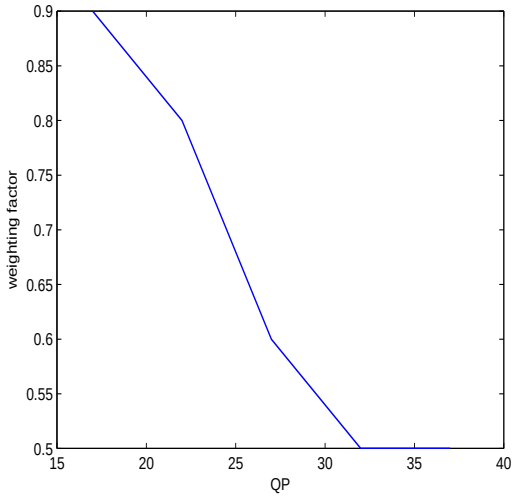


Figure 3-3. Average QP vs. weighting factor ω over all test sequences for intra only mode.

Table 3-1. ω value for different frame type and QP

QP	I	P	B
$QP <= 25$	0.9	0.5	0.5
$QP > 25$	0.5	0.5	0.5

over-filtering the major concern in s''. Hence, a more involvement of good quality s' helps to overcome over-filtering without loss to the filter output. When QP becomes large, the loss caused by low quality s' becomes noticeable and should be involved less in the filter input, as is suggested by the decrease of ω.

As mentioned before, we should consider to jointly design both deblocking filter and adaptive in-loop filter. To improve the overall performance, we observed that to increase the deblocking parameter α and β slightly will give better performance in relatively low QP range. Similar strategy has also been reported in [64]. In our algorithm, the better case of the two: with or without increasing deblocking parameter, is chosen at encoder based on the rate-distortion (RD) cost. A flag at slice header is used to signal the selected deblocking parameter setting if CQALF is enabled.

We summarize the analysis above in Table 3-1 which suggests the choice of ω that achieves the best coding performance.

3.3.3 Compatibility with QALF

As the estimated filter is optimal in the sense of minimizing the MSE of the whole frame, it may be far from optimal for each pixel. Actually, degradation can be observed for some pixels that are filtered by the estimated filter. To solve this issue, QALF introduces a region based signaling scheme to indicate whether a region is filtered or not. It adopts a bottom-up method for rate-distortion optimization (RDO) of quadtree data structure to signal the block partition and filtering decision [53].

We integrate our scheme on top of QALF framework to take advantage of the region based filtering scheme. Note that in CQALF, though the classification of the pixels makes more adaptive filter design and usually lower distortion than no classification, it sacrifices the bit rate by transmitting two sets of filter coefficients per frame. Consequently, for each frame, we compare the RD cost with and without classification, and select the one with a lower RD cost. If classification is enabled, it enables the proposed CQALF, otherwise, the algorithm goes back to QALF. The quadtree signaling scheme is adopted to indicate the partition and filtering decision, which is similar as that of QALF. Since the pixels are classified into two categories, for each block to be filtered, pixels within each category will be filtered by its corresponding filter.

3.3.4 Syntax and Semantics of CQALF

Table 3-2 gives the syntax of adaptive in-loop filter data supporting CQALF, which is sent in slice header syntax structure. The change is marked by the bold font. Refer to semantics below for specification of each new element. Note that if both of the two sets of coefficients exist, they are of the same tap size num_of_coeff_luma.

1) calf_on_flag: specifies the use of CQALF for the slice. calf_on_flag equals 1 indicates CQALF is applied for the slice. calf_on_flag equals 0 otherwise.

Table 3-2. Adaptive loop filter data syntax

adaptive_loopfilter_data{	Descriptor
pred_coef_mode	u(1)
cqalf_on_flag	u(1)
cqalf2_offset_2	u(1)
alf_tap_size_luma	ue(v)
for(i=0;i¡num_of_coeff_luma;i++) {	
filter_coeff_luma[i]	se(v)
}	
if (**cqalf_on_flag**==1)	
for(i=0;i¡num_of_coeff_luma;i++) {	
filter_coeff_luma2[i]	se(v)
}	
...	
}	

2) calf2_offset_2: specifies the use of deblocking parameter α and β. calf2_offset_2 equals 1 indicates α and β offset by 2. calf2_offset_2 equals 0 indicates no α and β offset.

3) filter_coeff_luma[i]:specifies filter coefficients for luminance in Category I.

4) filter_coeff_luma2[i]: specifies filter coefficients for luminance in Category II.

3.4 Experimental Results

In this section, intensive experiments are conducted to verify the performance of the proposed CQALF. Simulations are conducted using class B, C, D and E test sequences that are recommended by MPEG Call for Proposal. The test condition is based on [65]. KTA software version kta2.6r1 is used as the reference with all KTA tools enabled except for QALF when conducting the proposed CQALF tests. BD-PSNR is measured and compared according to [66].

We compare the BD bitrate saving of QALF and CQALF respectively over kta2.6r1 anchor without adaptive in-loop filter. We tested three coding structures, intra only, IPPP, and HB for the first 50 frames of each sequence. QP ranges from 22 to 37 with

a step of 5. Weighting factor ω is set according to Table 3-1. As shown in Table 3-3, CQALF always out-performances QALF in all three cases. We also demonstrate the rate distortion (RD) performance of CQALF and the anchor for two selected sequences: Kimono1_1920x1080_24 of IPPP coding structure in Figure 3-5, and vidyo1_1280x720_60 of HB coding structure in Figure 3-6.

Table 3-3. BD bitrate saving (%) of QALF & CQALF over anchor

Sequence	IIIII		IPPPPP		HB	
	QALF	CQALF	QALF	CQALF	QALF	CQALF
Kimono1_1920x1080_24	21.39	24.91	5.42	5.70	6.17	6.57
ParkScene_1920x1080_24	6.39	11.13	1.87	3.67	2.27	4.03
cactus_1920x1080_50	5.36	8.42	4.48	5.38	4.31	5.03
BasketballDrive_1920x1080_50	3.74	11.59	7.29	9.78	5.45	7.80
BQTerrace_1920x1080_60	3.58	7.04	5.22	6.43	7.61	8.49
BasketballDrill_832x480_50	10.30	11.96	10.06	11.96	5.69	7.90
BQMall_832x480_60	3.05	5.33	3.71	4.71	2.58	3.26
PartyScene_832x480_50	1.53	3.57	3.53	4.07	5.68	5.96
RaceHorses_832x480_30	5.68	7.99	2.18	2.39	1.55	1.83
BasketballPass_416x240_50	5.36	9.91	4.44	6.72	3.35	4.90
BQSquare_416x240_60	1.61	4.18	5.56	7.14	13.15	13.75
BlowingBubbles_416x240_50	1.94	3.32	2.48	3.32	2.89	3.31
RaceHorses_416x240_30	7.30	10.06	1.96	2.19	1.18	1.60
vidyo1_1280x720_60	8.54	11.99	8.92	20.53	5.06	6.76
vidyo3_1280x720_60	12.10	15.03	11.46	12.84	8.96	9.75
vidyo4_1280x720_60	7.25	11.33	10.37	14.02	6.78	8.02
Average	**6.57**	**10.05**	**5.56**	**7.55**	**5.17**	**6.19**

We observe that the coding gain of intra only case achieved by CQALF is the largest among all other cases. In intra frame, there are more pixels that may be modified by deblocking filter than in P and B frames. Hence the over filtering is more severe in I frame. As CQALF proposes to solve this problem, it can be foreseen that intra only case is the most effective to demonstrate the improvement of CQALF.

Figure 3-7 are selected partial frames extracted from two sequences to demonstrate the visual performance of CQALF. It can be seen the proposed CQALF achieves competitive or even better subjective quality than QALF.

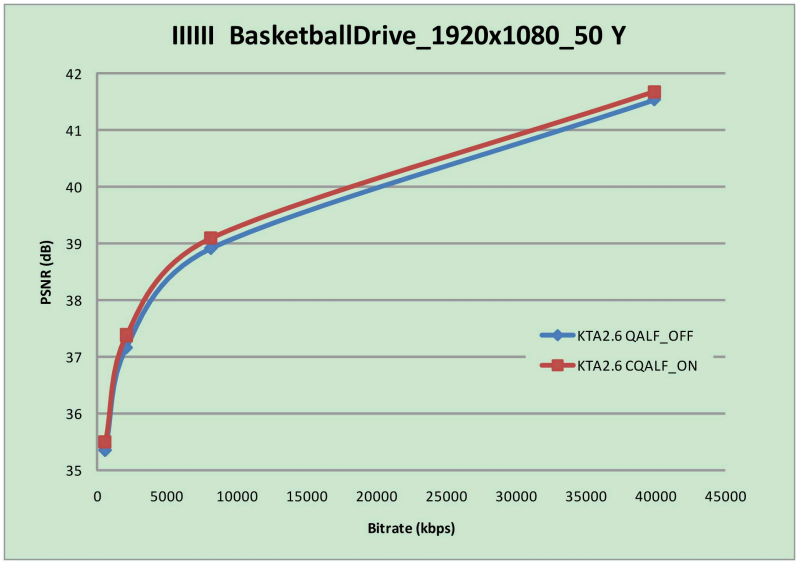


Figure 3-4. RD performance of CQALF vs. kta2.6r1 anchor of BasketballDrive_1920x1080_50 in intra only coding structure. BD bitrate saving 11.59%.

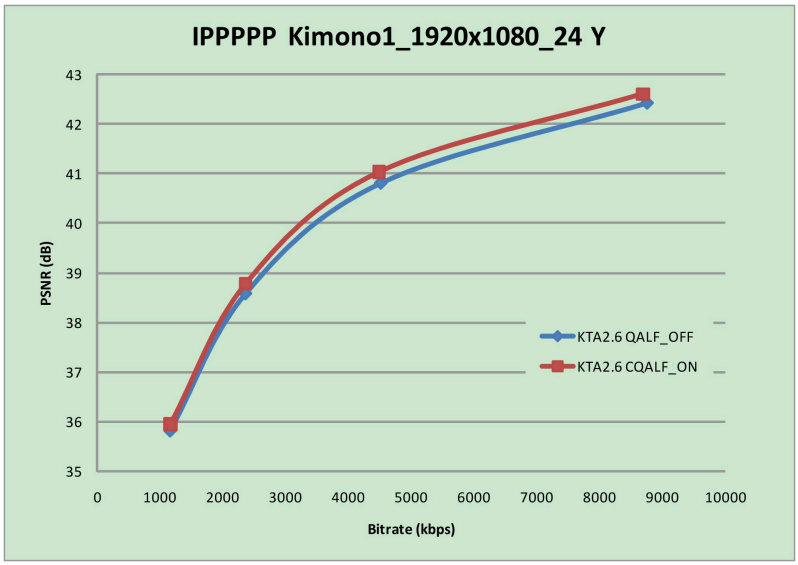


Figure 3-5. RD performance of CQALF vs. kta2.6r1 anchor of Kimono1_1920x1080_24 in IPPP coding structure. BD bitrate saving 5.7%.

3.5 Summary

We propose a classified quadtree-based adaptive loop filter to improve the coding efficiency by jointly considering the design of deblocking filter and adaptive in-loop filter (this work has been filed patent [67]). Pixels in a coded frame are classified into two

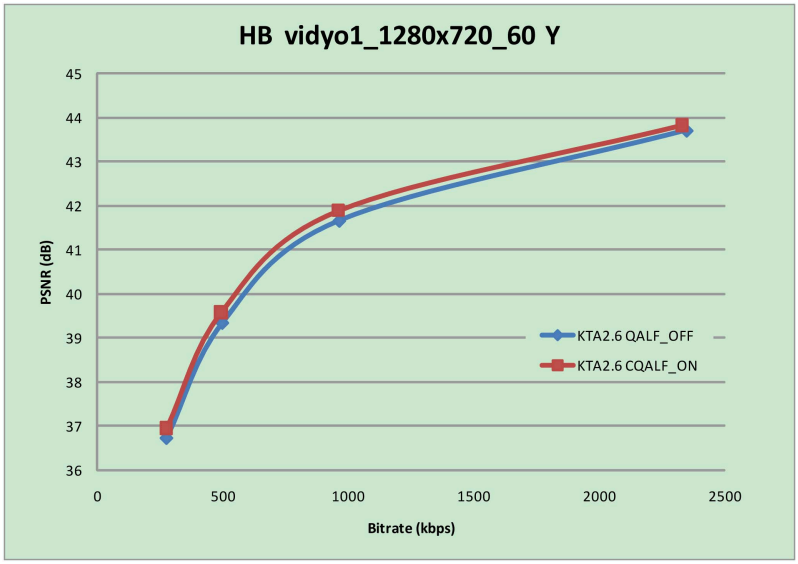


Figure 3-6. RD performance of CQALF vs. kta2.6r1 anchor of vidyo1_1280x720_60 in HB coding structure. BD bitrate saving 6.76%.

categories, pixels modified and not modified by the deblocking filter. A wiener filter is carefully designed for each category. For the pixels modified by the deblocking filter, the reconstructed pixels before and after deblocking filter are combined as the input of the filter. For the pixels not modified by the deblocking filter, only reconstructed pixel after deblocking filter is used. Extensive experiments are conducted to verify the performance of the proposed CQALF. Compared with kta2.6r1 anchor, CQALF provides an average of 10.05% bitrate deduction for intra only, 7.55% for IPPP and 6.19% for HB coding structure.

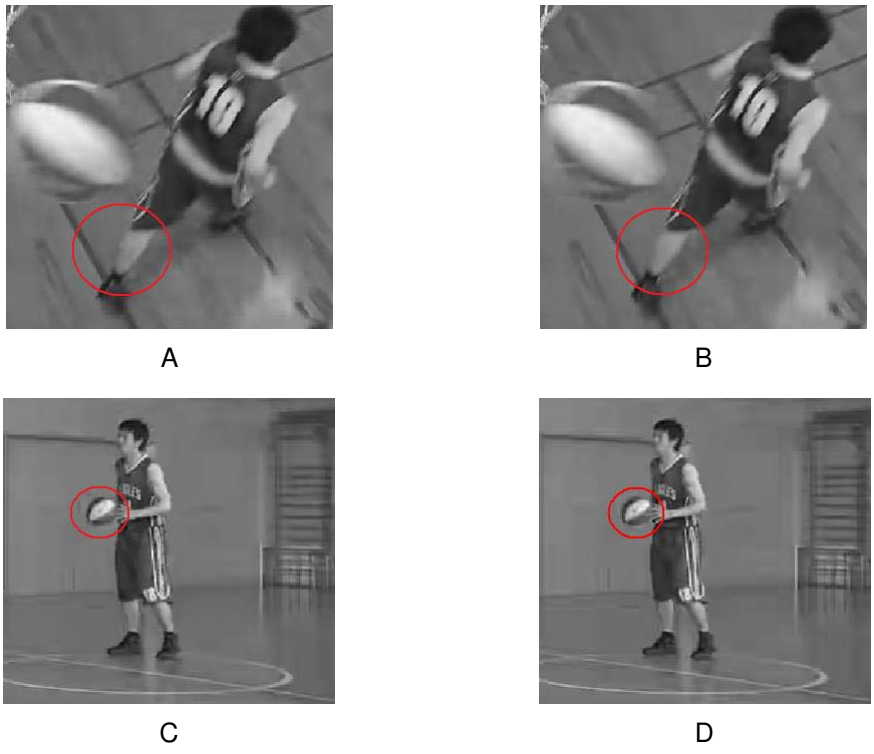


Figure 3-7. Reconstructed BasketballDrill (832x480) sequence at the 10th frame with QP=37. (a) QALF: 32.12dB; (b) CQALF: 32.72dB; Reconstructed Basketballpass (416x240) sequence at the 9th frame with QP=37. (c)QALF: 31.96dB; (d)CQALF: 32.00dB

CHAPTER 4
A ROBUST VIDEO HASH SCHEME BASED ON 2D-DCT TMO

In this chapter, we propose a video hash scheme that utilizes image hash and spatio-temporal information contained in video to generate video hash. Experiment results show that the proposed 2D-DCT temporal maximum occurrence (2D-DCT TMO) scheme successfully derives shot hash that represents the content, and is very robust in video identification, authentication, and verification.

4.1 Research Background

Video hash techniques, also called video signature or fingerprinting, extract the most important features from a video to form compact digests that allow efficient visual content identification and authentication. Existing approaches of video hash can be classified into two categories. 1) Schemes in the first category extract features in spatial domain only. Usually, a robust image hash approach is first developed in this kind. Video hash is a simple extension of the image hash applied to those individual *key* frames extracted from the video [18][19]. 2) Schemes in the second category extract features in spatio-temporal domain. The scheme in [68] applies 3-Dimensional Discrete Cosine Transform (3D-DCT) transform to a normalized video clip, and selects a $4 \times 4 \times 4$ cube of coefficients as video hash. Since 3D-DCT contains both temporal and spatial information, this is a video hash scheme based on spatio-temporal domain. In [69], each incoming frame of a video clip is divided into small blocks. The mean luminance is calculated for each block. The hash value is the sign of the spatio-temporal difference of mean luminance. Another hash technique based on spatio-temporal color difference is reported in [70]. Each frame is divided to small blocks, and the difference between spatial and temporal similarity is used as hash function, where similarity is defined as correlation between spatial/temporal consecutive blocks.

In the spatial domain video hash technique, how to identify *key* frames to efficiently represent a video sequence is the most important issue. Intuitively, whatever key

frame selection method is employed, key frames are only selective information of the video in both temporal and spatial domain, and they can never truly be a good global representative of the video sequence. For example, though Radial hASH works well on still images, it is not sufficiently robust in discriminating videos if simply using the image hash of key frames as video hash [18]. Meanwhile, video hash techniques in the second category extract features in spatio-temporal domain, which are completely independent solutions from key-frame based methods, as they are based on image hashing. Generally, approaches employing both spatial and temporal information tend to be more robust than key-frame based schemes because the features extracted from 3D domain are surely a better representation than 2D domain. Unfortunately, approaches reported in this category are very limited compared with the well-developed image hashing techniques [18][71][72][73][74]. Hence, we are motivated to find a better way to integrate image hash to video hash than simply applying it to some selected key frames. Besides, though 3D-DCT based video hash [68] is robust, it requires pre-processing to normalize video in spatial and temporal as to convert it to a standard cube before applying 3D-DCT. Temporal smoothing may remove minute variations of a pixel in time, but for variable length video sequences, if all temporally compressed to fixed frame length, say $F = 64$, will harm the representation of the video, making it more difficult to extract meaningful features in video hash. Also, this method is very sensitive to the play order of the video, as reverse play temporally affects the DCT coefficients. Therefore, we are interested to improve 3D-DCT based video hash by solving the problems.

We propose a robust video hash scheme based on 2D-DCT temporal maximum occurrence (2D-DCT TMO). It extracts spatio-temporal information. We observe that to extract features directly from a complete video sequence is very difficult. While a video shot, i.e., a group of frames of continuous action in time and space, is a more meaningful unit for video processing and identification. Therefore, we first divide a video sequence to multiple shots. We then apply 2D-DCT to each frame in a shot, quantize

the DCT coefficient, and record the temporal occurrence of the co-located coefficient. We select those DCT coefficients of the maximum occurrence and do inverse DCT transform back to spatial domain to construct a representative frame, and calculate the image hash for this frame, set the frame hash as video hash for this shot. Hence, video hash is actually the *shot hash* in our scheme.

The rest of the paper is organized as follows. In section 4.2, some basic procedures before deriving video hash in 2D-DCT TMO is introduced, including shot segmentation and DCT coefficient selection. Section 4.3 gives details of 2D-DCT TMO, explaining how to obtain shot hash in pairwise manner to 2D-DCT maximum occurrence. Section 4.4 presents the experimental results and discusses its performance. And section 3.5 concludes the paper.

4.2 Some Basic Procedures Before 2D-DCT TMO

In our method, a video sequence is first divided into several video shots. Each shot consists of a group of frames of continuous action in time and space. And then we spatially normalize shot to a pre-defined WxH size. Unlike in [68], we do not temporally normalize a video sequence to V(W,H,F) 3D cube, where (W,H) and F are respective size on spatial and temporal dimensions after subsampling. On the other hand, we also do not do temporal normalization to the segmented video shot. Because we propose to count the number of temporal occurrence of spatial features (2D-DCT histogram), and each frame will be a valuable sample in statistics.

4.2.1 Video Shot Segmentation

The goal of video shot segmentation (video shot detection) is to find significant disparities between consecutive frames of the sequence. In our scheme, we want to localize the shot boundary as accurate as possible so that all frames in a shot display similar temporal and spatial features. Also we want the shot segmentation to be stable, i.e., the detected shot boundary will not be affected by video length. Any video shot detection method that meets the requirements above can be used

for shot segmentation. In this paper, we employ a shot detection based on regional histogram difference [75]. A frame is divided to certain number of blocks, say $4 \times 4 = 16$ blocks, compute the gray level histogram difference between collocated blocks of two consecutive frames, and count those blocks with the difference metric greater than a certain threshold TH_1. If the number of such blocks exceeds a pre-defined value TH_2, we decide this frame is a shot boundary frame. From our experiment, we find $TH_1 = 0.1$ and $TH_2 = 12$ shows quite good performance.

4.2.2 DCT Coefficient Selection

The DCT transforms an image from spatial to frequency domain. As most of the energy is now concentrated in low-frequency region, it forms a much more compact representation of the image than in spatial pixel domain. That is, the amount of information contained in a few DCT coefficients may be close to all $M \times N$ pixel values. Hence, we choose 2D-DCT coefficients as the features extracted from spatial domain. As mentioned before, we normalize each frame to a pre-defined spatial scale, which we choose to be QCIF (176) size in this paper. Then we need to figure out how many 2D-DCT coefficients are enough for a QCIF size image. Figure 4-1 shows a typical DCT energy spectrum, i.e., DCT variance distribution along with the DCT index in zig-zag scan order. It is based on a randomly picked video clip from *Big Bunny Buck* of 5000 frame length. Obviously, variance, as an energy metric, decreases sharply as DCT index grows. While small index number indicates low frequency component, and high index for high frequency, which can be guaranteed in zig-zag scan order. Figure 4-2 shows in a more straightforward way of energy distribution over DCT index(frequency component). The 3 curves in the figure represents 3 randomly picked video clips extracted from *Big Bunny Buck* of 5000 frame length each. They are of the similar shape. It is seen that most of the energy compacts in the first few DCT coefficients (low frequency components) in DCT domain. Figure 4-3 gives a closer look at the energy ratio of the first 100 DCT coefficients. As a tradeoff between accuracy and complexity, we select the

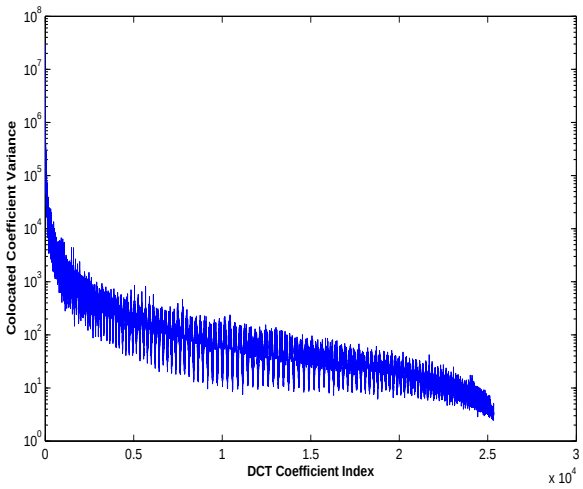


Figure 4-1. DCT Energy Spectrum

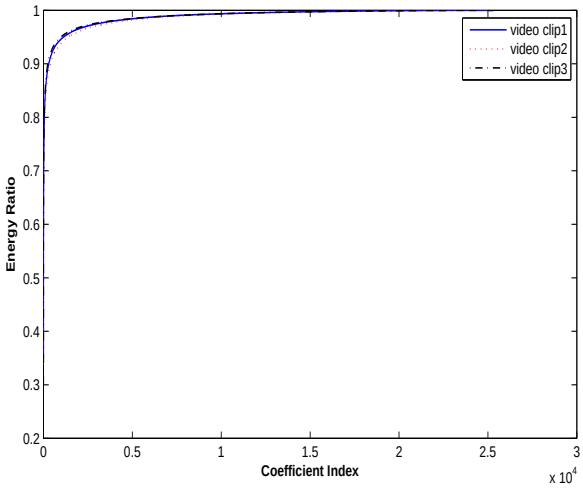


Figure 4-2. Energy Ratio with the Increased DCT Coefficients.

first 28 DCT coefficients in zig-zag order, which takes up around 70% of all energy, as the extracted spatial feature. This number is also used in [76] without explanation.

4.3 Shot Hash Based on 2D-DCT TMO

The first 28 DCT coefficients in DCT domain of each frame are good spatial features. As all video frames in a video shot have similar visual content, the collocated DCT coefficients along the temporal order will also show similar values, and a value or values (if several values occur in equal frequency) that are most likely to occur can reflect the features of the entire video shot. To derive the temporal feature of a video

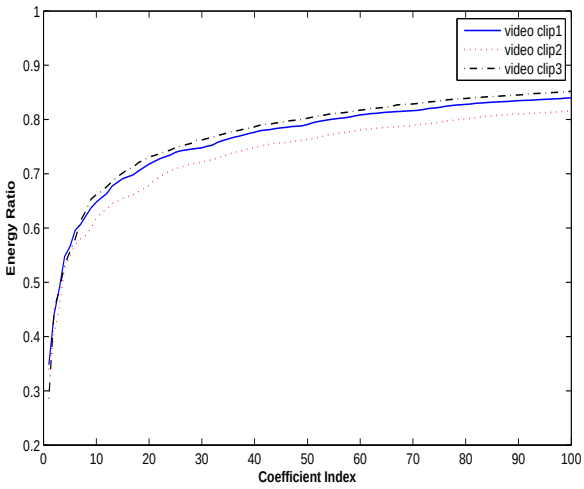


Figure 4-3. Energy Ratio with the First 100 DCT Coefficients.

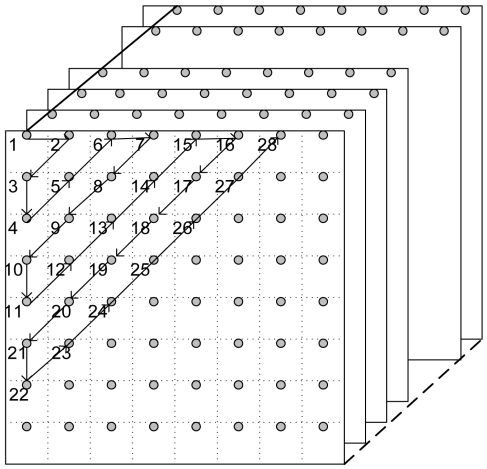


Figure 4-4. Zig-Zag scan order for the selected 28 DCT coefficients.

shot, we count the number of occurrence of the collocated DCT coefficients in a video shot in a histogram manner. For example, assume a video shot has T frames, and all first 28 DCT coefficients are carefully quantized. Among T DCT coefficients marked Number 1 in zig-zag order, we do histogram statistic, and re-order the histogram in descent order to find the DCT value/values most likely to occur. Likewise, we copy this procedure to all remaining 27 DCT coefficients in these T frames. This procedure has been demonstrated in Figure 4-4.

Intuitively, for each of the 28 DCT coefficient entry, a single value that occurs most frequently should be able to represent the entire video shot. However, we cannot simply pick one most frequently occurring value, due to the following reasons:

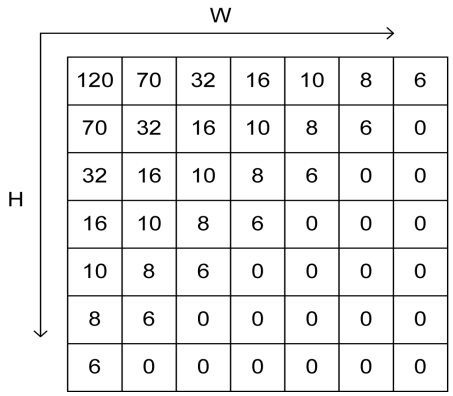
1) If there are several values of equal occurrence, to pick any one of them will inevitably lose the accuracy of temporal information. No single value has priority over others if they all appear the same times given the frame number.

2) Because of the post-processing to video shot, such as geometric manipulation, temporal manipulation and compression etc., the one DCT value having most occurrence in such video shot may differs from the true value in the original video shot. How much it differs from the true value depends on the process itself, but from our experiment, such difference does exist in most of the video shots inspected.

3) The importance of DCT coefficients differs. Low frequency components contain more energy and information [76], and should be assigned a larger number of values to choose from.

To this end, we design a quantization table for 28 DCT coefficient entries based on the one in [76] as in Figure 4-5. The number in each entry is the quantization level for that DCT coefficient, for instance, the first DCT is equally quantized to 120 levels, the second and third DCT are quantized to 70 levels, etc. Note that the smaller DCT index, the greater quantization level reserved for that entry, which indicates the importance of each DCT coefficient.

As mentioned in Section 4.1, the application of video hash is efficient visual content identification, video database search etc. These applications have one thing in common - it involves at least two video contents, one suspect video and one reference video. All the existing video hash methods derive video hash independently for both videos and measure the distance between the two hashes to decide the content similarity of the two videos. It guarantees there is one unique hash for each video. Instead, in this paper,



120	70	32	16	10	8	6
70	32	16	10	8	6	0
32	16	10	8	6	0	0
16	10	8	6	0	0	0
10	8	6	0	0	0	0
8	6	0	0	0	0	0
6	0	0	0	0	0	0

Figure 4-5. Quantization Table of the selected 28 DCT coefficients.

we propose a novel scheme that derives video hash in a pairwise manner, i.e., no hash for an individual video can be obtained without referring to another video. Obviously, this breaks the unique property of the video hash, since different video hashes will be derived from the same video content referring to different references. However, the proposed pairwise manner shows excellent performance in all applications in Section 4.4. Figure 4-6 illustrates the flow chart of the pairwise shot hashes derivation. After quantizing each DCT entry based on the quantization table and reorder them in descending order for the individual shot, we need to collaboratively select a pair of coefficients for each collocated DCT entry from the reserved candidate levels in a shot pair. A simple way is to always choose a pair of collocated DCT coefficients that are closest to each other. Regarding that closer DCT values indicate more similarity in spatial domain, we guarantee that the pairwise shot hashes always indicate the most content similarity. Hence, if the distance metric between such two shot hashes is still large, they are very likely to have completely different contents.

Furthermore, we point out that to choose the closest value DCT pair may not always be a good choice. For example, if there is an exact same DCT value in collocated DCT entry in both shots, but in reference shot it occurs very frequently, while in suspect shot it is very rare, such DCT value should be excluded from the candidate level since we prefer DCT levels with high temporal occurrence in both shots. Therefore, among all

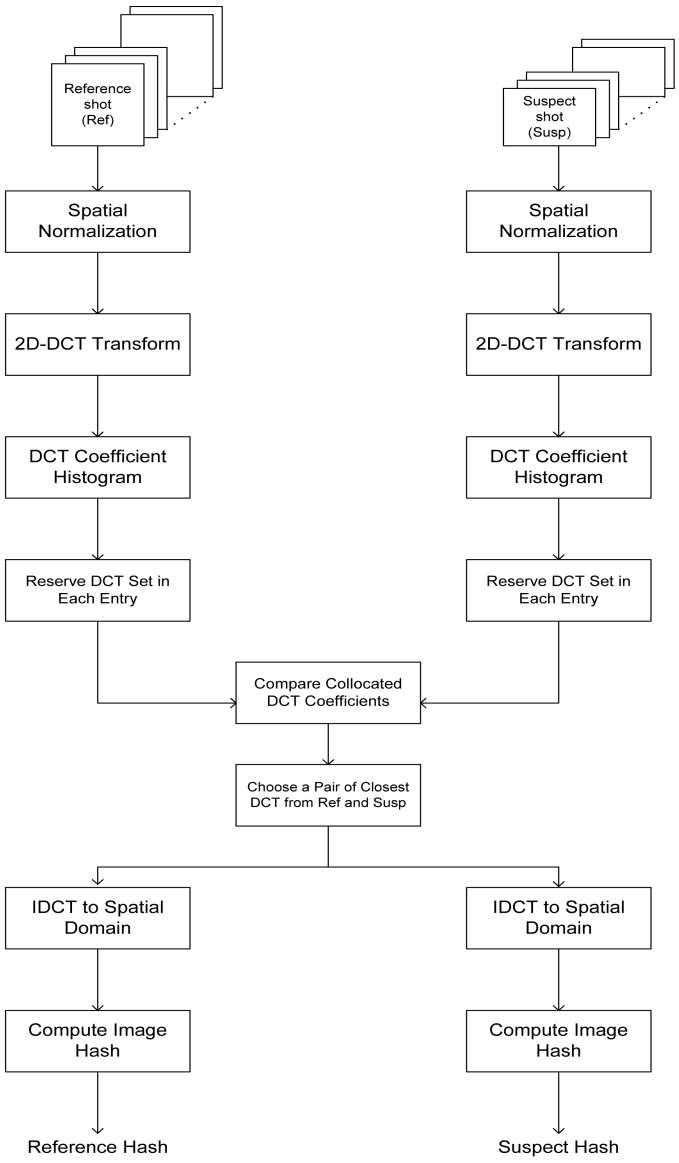


Figure 4-6. Flow chart of deriving pairwise shot hashes from two videos.

the quantization levels in descending occurrence order for each DCT entry (a frequency spectrum), only those taking up the first half of the energy in the frequency spectrum will be reserved as candidates. We do such screening process for both shots so that only good temporal representative DCT levels are kept and selected in pairwise manner for shot hashes. Figure 4-7 shows the frequency spectra of the first 3 DCT entries after reordered quantization for reference shot and suspect shot respectively (the 5th shot from *Big Bunny Buck*). When restricting the selection range within the first half energy

Table 4-1. Number of DCT levels kept for the first 3 DCT entries in reference and suspected shots

	DCT 1	DCT 2	DCT 3
Reference shot	3	3	2
Suspected shot	3	3	2

of each spectrum, only a few DCT levels of the highest temporal occurrence will be kept, as shown in Table 4-1. We then choose a pair of DCT levels of closest values for each collocated DCT entry. Finally, we have a pair of 28 DCT coefficients for both reference and suspect shots.

In our proposed 2D-DCT TMO scheme, the 28 DCT coefficients are not the final shot hash. We have to inverse transform the DCT coefficients and derive image hash from the spatial domain as the ultimate shot hash. It is termed in this paper as *feature frame* for a shot. This is based on the following considerations:

1) The structure of the 28 DCT coefficients between any two shots is quite similar, with one large value of the first DCT coefficient (DCT 1) and the rest are much smaller values compared to DCT 1. This structure makes it difficult to distinguish two shot hashes because the one large value of DCT 1 makes the others all look like 0.

2) As mentioned before, we are motivated to find a better way to integrate the well developed image hash methods to video hash.

It is expected that the more similarity the two shots have in content, the more visually alike of the spatial pictures. Given the fact that the distance between image hashes depends on the difference of the image contents, we can use the image hash of the feature frame as the shot hash. Any image hash method that is unique and robust can be implemented. In this paper, we employ the Radial hASH in [18], and compute the cross correlation of feature frame hashes as the distance measure.

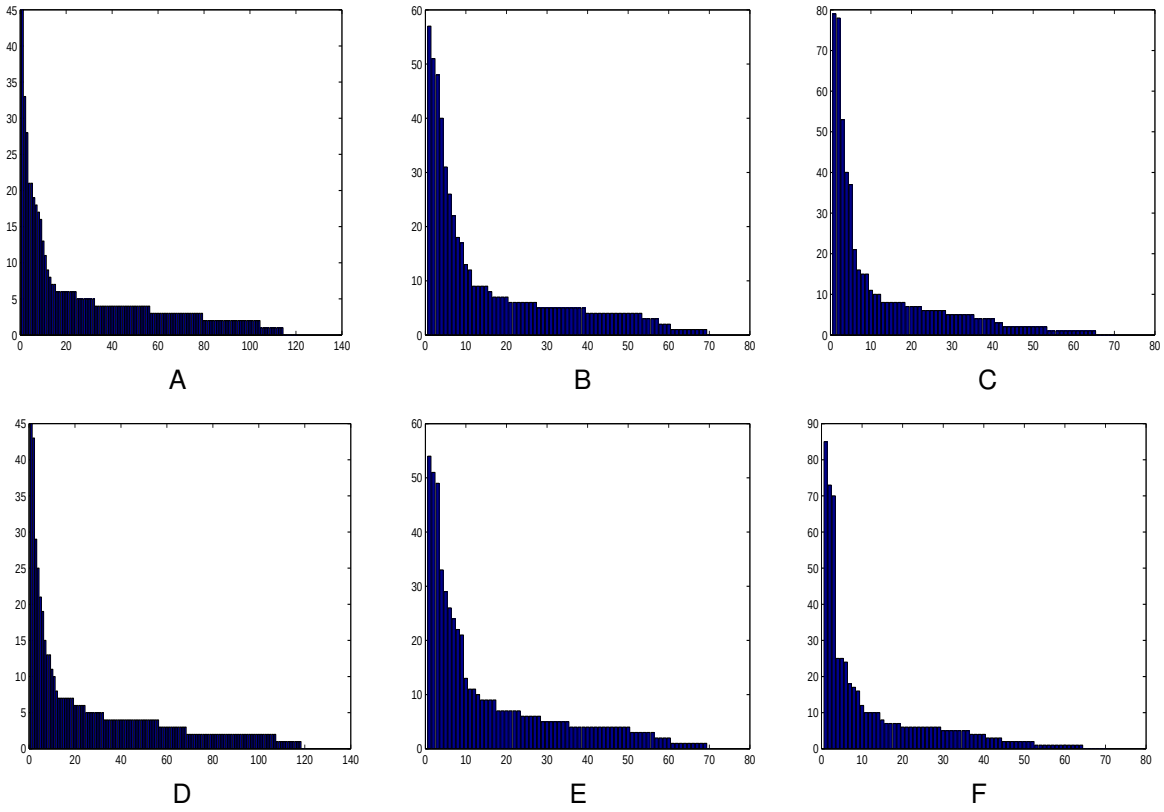


Figure 4-7. Frequency spectra of the first 3 DCT entries for reference shot (original video) and suspected shot (compressed video). (a) Reference shot: DCT 1 (b) Reference shot: DCT 2 (c) Reference shot: DCT 3 (d) Suspected shot: DCT 1 (e) Suspected shot: DCT 2 (f) Suspected shot: DCT 3

4.4 Experimental Results

In this section, we report the identification and verification performance of the proposed 2D-DCT TMO video hash scheme.

The identification problem is defined as the ability to recognize a video shot (reference shot) from video clip database (collection of suspect shots). What makes it difficult is the suspect shot may have been attacked or modified and differs from the true version in reference.

In our experiment, we take a 10 min length video clip from movie *Big Buck Bunny* as the test video. Certain shots are extracted as reference. Suspect video, which is the full 10 min length video, will firstly be segmented to shots, and we try to identify the

Table 4-2. Partial Cross Correlation Results from Identification Experiment (R: Reference shot number, S: Suspected shot number)

	S14	S16	S22	S42	S44
R14	**1.00**	0.9994	-0.6126	0.4759	0.4876
R16	0.9994	**1.00**	-0.6241	0.4526	0.4685
R22	-0.6126	-0.6241	**1.00**	-0.1215	-0.1571
R42	0.4759	0.4526	-0.1215	**1.00**	0.9990
R44	0.4876	0.4685	-0.1571	0.9990	**1.00**

reference shots from the the entire suspect video clip. For a given reference shot, we compute the shot hash in a pairwise manner with each shot in the suspect video clip, and identify the one with the highest cross correlation value as the match shot. Table 4-2 presents partial results of the identification experiment. We notice that cross correlation peaks at exactly 1 if the reference and suspect shots do have the same content. Figure 4-8 gives the feature frames of the reference shot and its matching suspect shot of Shot 14 and Shot 22. It can be seen that they are exactly the same, and therefore the image hashes derived from such frames should be identical. Meanwhile, we find the cross correlations between two shot pairs, Shot 14 & Shot 16 and Shot 42 & Shot 44 are very close to 1. Likewise, Figure 4-9 shows the feature frames of the two pairs, which are highly alike visually. And we also observe that the contents of the two shots, including the figure and the background of the shot pair does appear very high similarity, which are shown in Figure 4-10. Figure 4-10A and Figure 4-10B are frames extracted from shot 14 and 16 respectively, while Figure 4-10C and Figure 4-10D are from shot 42 and 44. Consequently, our proposed 2D-DCT TMO can successfully identify shot from the database, and the derived shot hash does represent the content of the shot. The more content similarity of the shot pair, the higher cross correlation between shot hashes.

The verification problem, on the other hand, is defined as the effort to prove or disprove if a video clip is what it is claimed to be. The suspect video usually is altered, post-processed or modified in various ways before the verification test. Hence, a robust video hash is necessary in order to tolerate the malicious attacks or accidental

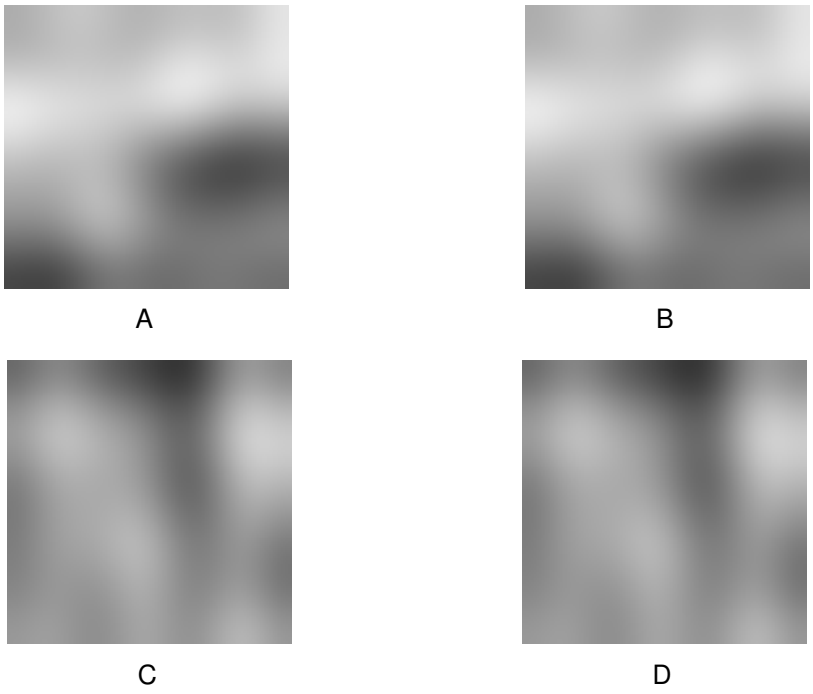


Figure 4-8. Feature frames for the reference shot and its matching suspected shot of Shot 14 and Shot 22. (a) Feature frame for reference shot 14 (b) Feature frame for suspected shot 14 (c) Feature frame for reference shot 22 (d) Feature frame for suspected shot 22

manipulations to the video. These changes include but not subjected to geometric transform, blurring, frame drop, time clipping, compression and noise interference. The performance of verification depends on the cross correlation between the shot hashes of the suspect shot and the reference shot. A predefined threshold of cross correlation is set to decide either to accept or reject the claimed content. In our experiment, we purposely manipulate an original shot content (Shot 22 in *Big Buck Bunny*) in different ways to create various versions of suspect shots, and compare these shot hashes with the reference shot hash of the original content. These manipulations are geometric transform (shift, rotate and resize), frame drop (periodically shrink to 1.5 times shorter of the original length), time clipping (cut shot to 40% original length), contrast change, blurring, through AWGN channel, and video compression (H.264/AVC). The result is shown in Table 4-3. Note that the cross correlation between the shot hashes of the

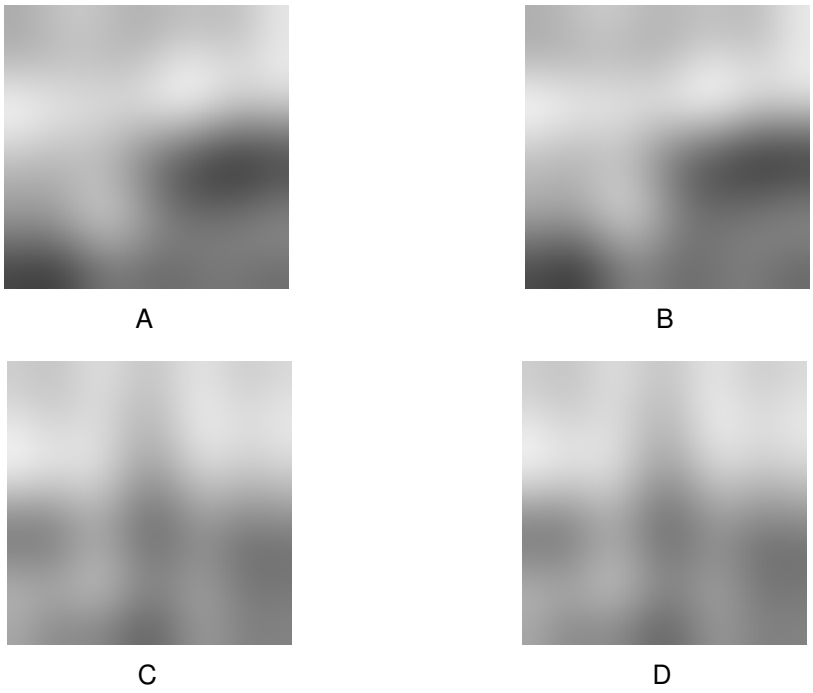


Figure 4-9. Feature frames for two non-matching shot pairs with high cross correlation. (a) Feature frame for reference shot 14 (b) Feature frame for suspected shot 16 (c) Feature frame for reference shot 42 (d) Feature frame for suspected shot 44

Figure 4-10. Extracted frames from two shot pairs that appear similar content in *Big Buck Bunny*. (a) Frames from Shot 14 (b) Frames from Shot 16 (c) Frames from Shot 42 (d) Frames from Shot 44

Table 4-3. Verification Performance

Methods of modification	Cross correlation
Shift(15 pixels)	0.9759
Rotate(20 pixels)	0.9989
Resize(10 pixels)	0.9939
Frame drop	0.9999
Time clipping	0.9994
Contrast change	0.9938
Blurring	1.00
AWGN	0.9999
Compression	0.9999

altered version and the original shot is no lower than 0.97, which verifies the robustness of the proposed hash scheme. In fact, extensive tests over sequences all show the similar result. Therefore, we conclude that 0.97 is a good threshold to decide whether two shots are visually similar or not.

4.5 Summary

In this chapter, we proposed a robust video hash scheme based on 2D-DCT temporal maximum occurrence. The video clip is firstly segmented to shots, and video hash is actually derived in unit of shot. We notice that for video hash applications in identification and verification, reference video and suspect video appear in pair. Therefore, we propose to derive the shot hash in a pairwise manner. For both reference and suspect shot, we count the number of quantized DCT coefficients in each of the 28 selected DCT entry, only keep those of the highest temporal occurrence, and choose a pair of the closet value as the DCT coefficient for every collocated entry. We then do inverse transform to the reserved DCT coefficients for both shots, and derive image hashes from two feature frames by Radial hASH. Experiment results show that the proposed 2D-DCT TMO successfully derives shot hash that represents the content, and is very robust in video identification, authentication, and verification applications.

CHAPTER 5
DELAY-RATE-DISTORTION MODEL OF SOURCE VIDEO CODING

Video communication over wireless networks has experienced extensive growth in the past decades. However, the QoS support in the wireless network, particularly end-to-end distortion guarantee to enable real time video application is still a valid research problem. In this paper, we briefly review the current research on delay constrained video communication system. We point out that all existing works assume encoding time a constant and derive relationship between transmission delay and system performance.

In this chapter, we propose a delay-Rate-Distortion (d-R-D) model in source video coding to adapt encoding time, source distortion and source rate for both real time video coding as in IPPPP mode and none real time video coding as in hierarchical B pictures. In real time video coding, with d-R-D source coding model, we are able to vary transmission delay by tuning encoding parameter to change encoding time, given fixed end-to-end delay. Ref [77] briefly analyze the transmission performance with variable transmission delay based on a Markov chain model, which can be translated to system throughput versus encoding buffer size. However, due to the true packet arrival rate is not constant as assumed in Markov chain model, it is difficult to accurately model the entire end-to-end video communication system and estimate the packet transmission error over wireless channel given d-R-D source model. Hence, we analyze the relationship between packet transmission error rate and encoding time through simulation, to prove the fact that the control of encoding time will influence the end-to-end system performance. For non real time video application, where B frame is allowed to get better coding efficiency, we establish the d-R-D source coding model for hierarchical B pictures coding. In hierarchical B pictures, encoding time is largely dependent on the number of hierarchical B pictures level or group of picture (GOP) size. Since delay constraint is not vital in non real time coding applications, we only relate

rate distortion performance of source coding with GOP size, without further analysis of end-to-end system performance.

5.1 Model of Source Video Coding in Real Time Coding

5.1.1 End-to-end Delay in Video Communication System

With the development of wireless communication and dramatically increase of mobile smart phone use,video streaming over wireless network becomes a big concern in both industry and academics. Examples of such applications are IPTV over mobile phones and sharing over social network via smart phones. To support these applications, future wireless networks are expected to provide quality of service (QoS) guarantees, including end-to-end delay, data transmission rate, and packet error probability. The QoS requirements pose great challenges as wireless fading channels may cause severe QoS violations.

For real time video applications, the system has a maximum tolerable end-to-end delay for each video frame, from the moment the frame captured at the encoder to the identical frame display at the decoder. Due to the fading and multi-path interference of wireless network, the packet transmission error is inevitable. Therefore, certain error correction scheme needs to be employed to protect packet in error-prone channels. Forward Error Correction (FEC) coding adds redundancy among all packets for packet protection. Automatic Repeat Request (ARQ) will cause unbounded delay, and not appropriate for real time application. However, ARQ feedback combined with error control has been proved to be an alternative scheme to FEC[78][79]. An architecture employing hybrid ARQ with FEC is given in [80] for low delay wireless video transmission.

All research on wireless video transmission with delay constraint have demonstrated the relationship between delay and system performance, such as distortion, packet loss rate and throughput[81]. [78] minimizes the source coding distortion with the rate constrain transformed from delay constraint. [79] formulates the problem as to minimize

the expected end-to-end distortion with the delay constraint. [82] maximizes the packet retransmission time with delay constraint by adaptive packetization and scheduling scheme. [83] maximizes the end-to-end distortion with wireless network delay. [77][84] analyzes the system throughput with delay constraint, which is equivalent to minimizing packet error probability with a bounded buffer length. Analytical and simulation results show that delay has tremendous effect on system performance, and usually system performance is a non-monotonous decrease function with delay. How to balance the delay and performance by tuning system parameters is still a valid research problem.

Up to now, the analysis of delay in wireless video transmission can be categorized to three different methods. 1) The wireless video communication is modeled as a point-to-point system, and the end-to-end delay comes from the waiting time in encoder and decoder buffer, and the channel delay, including propagation delay and retransmission time (ARQ). It translates delay constraint in real time video transmission into a rate constraint, where the applicable rate constraints depend on future channel rates. Hence the delay requirement is equivalently to a rate control problem at the encoder. Examples in this category are [78][79][82]. 2) It is also a point-to-point system model. The delay mainly comes from the waiting time in the buffer at link layer of finite length. The delay will affect packet drop probability and delay bound violation probability. And packet error rate of the communication system is the sum of packet transmission error rate over fading channel, packet drop rate from buffer overflow and delay bound violation rate due to maximum packet delay constraint. [77] derives each of the rate and minimizes the packet error rate, which is equivalent to maximize the system throughput with delay constraint. 3) It models the video transmission over wireless networks, and the delay is due to packet transmission in a multi-hop network [83]. The end-to-end video distortion is the sum of source coding distortion and transmission distortion. It exploits a distortion-aware wireless video scheduling scheme and derives a bound on the asymptotic decay rate of system distortion. However, the source coding distortion

model in [83] is too rough and not be able to get an accurate analysis of delay-distortion balance.

The definition of delay in video communication depends on different applications. In real time video communication such as video conference, delay means the amount of time between the action from one party and the other party see the action. In broadcast video, delay means the amount of time between you press the button on your remote control to switch to a new channel and the new channel shows up on your TV. The sources of delay in video communication system includes but not restricted to the following[85]:

1) The time between an action and the camera shutter stops to capture he action;

2) The time to dump the picture from camera to devices;

3) The time for frame pre-process before feeding to encoder;

4) The time to re-order the picture before encoding if the encoding mode includes B frames;

5) The encoding time that compresses a picture frame to binary bitstream;

6) The encoder buffer delay used to support rate control and channel control;

7) The time it takes to multiplex or packetize the video and perform channel coding;

8) The transmission delay over the channel;

9) The time it takes to perform channel decoding and demultiplex or depacketize the video;

10) The decoder buffer delay to re-order the received picture if erroneous transmission happens or decoding includes B frame;

11) The decoder time that decompress a bitstream to a picture frame;

12) The time for frame post-process before display;

13) Other time delay added to allow synchronization to other events like audio sync.

In this chapter, we define the end-to-end delay in a wireless video communication system as from the moment the frame is fed to the encoder at the transmitter to the time

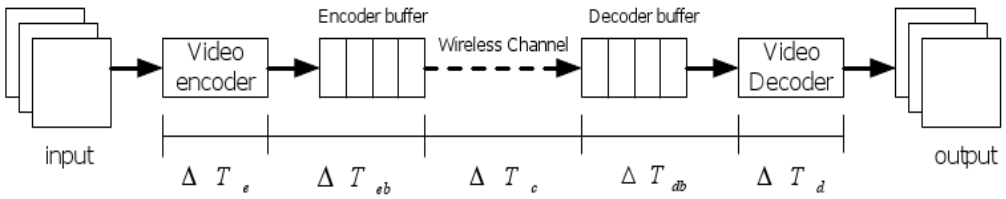


Figure 5-1. End-to-end delay of a wireless communication system.

it is displayed at the receiver. It is composed of the following: video encoding time ΔT_e, encoder buffer queuing time ΔT_{eb}, wireless channel transmission time ΔT_c, decoder buffer queuing time ΔT_{db} and decoding time ΔT_d [79][86], as shown in Figure 5-1. Note that we neglect several delay components to simplify the model. This definition matches the one in real-time application of video conference, where the delay constraint is so strict that typically no B frame is allowed in encoding mode. In this section, we use IPPPP coding mode as the base on which we develop the d-R-D model of real time video coding.

Each video frame is encoded and packetized to several packets for transmission. In real time application, the maximum end-to-end delay that each frame experiences is a constant, i.e.

$$\Delta T_e + \Delta T_{eb} + \Delta T_c + \Delta T_{db} + \Delta T_d = C \tag{5–1}$$

All previous studies have assumed encoding time and decoding time is of much smaller variation than the buffer delay and transmission delay, hence ΔT_e and ΔT_d are assumed constant and neglected in delay analysis. Therefore, they only focus on the system performance with buffer delay and transmission delay. However, the encoding time can be mapped to encoding complexity, and encoding complexity will affect source distortion [87]. If end-to-end delay is fixed, by tuning encoding time, the transmission distortion will also be affected as the buffer and transmission delay change. Hence, the overall system performance depends on how to assign end-to-end delay among different

time components. And we are seeking the optimal time arrangement among encoding and the rest of the delay to maximize the system performance given end-to-end delay.

To achieve this, we need a model that relates encoding time with source distortion and bit-rate. However, there is no such model in literature to the best of our knowledge.

5.1.2 Model of Source Coding in IPPPP Mode

In this section, unless otherwise explicitly specified, "coding" indicates "source coding". For Delay-Rate-Distortion model of real time coding, we want to model coding time (delay), coding rate and distortion as functions of some coding complexity parameter. We use divide-and-conquer scheme to segment encoding complexity to 3 parts: motion estimation, PRE-coding (include transform, quantization, inverse quantization and inverse transform) and entropy coding. This segment method is introduced for H.263 encoder in [88]. Motion estimation complexity is characterized by the number of sum-of-absolute-difference (SAD) operations, and PRE-coding complexity depends on the number of non-zero Microblock (MB) in a frame, and entropy coding complexity is characterized by coding rate. The entire coding time is a function of all three parts, and a rate-distortion model is derived for each part as a function of the complexity parameter of that part. However, the distortion vs. SAD model in H.263 does not apply to H.264/AVC. Hence, we discard the model in [88] and derive a new rate-distortion model as function of other coding parameter.

In Li's work [89], distortion and rate are derived as functions of the variance of prediction residue σ and quantization step size Q. This means distortion in inter frame is determined by two things: 1)Motion estimation (ME). The prediction residue data is a result of motion estimation. In this paper, residue data is assumed to be Laplacian distribution, and σ is the variance of the distribution. 2)Quantization. The quantization step size Q determines quantization distortion given prediction residue. And the overall encoding distortion is a combination of the two factors. Generally, a more complete motion estimation gets smaller residue, and a smaller quantization step size gets

smaller quantization loss. When Q is fixed, distortion of inter frame depends on ME, which can be characterized by three main coding parameters, MB coding mode, ME search range λ and number of reference frame θ. If we disable intra coding mode, the block coding for inter frame chooses from one of the following modes in H.264/AVC: skip mode, 16×16, 16×8, 8×16, 8×8, 8×4, 4×8 and 4×4. If given only one coding mode for a inter frame, the number of SAD operation for each block is

$$\#SAD = \lambda \times \theta \tag{5–2}$$

If we can model σ as a function of λ and θ, based on R-D model in [89], distortion D and bit-rate R can be written as functions of λ and θ. In the mean time, delay d is a function of encoding complexity parameters and power P. Note that σ is independent of PRE-coding and entropy coding. If we can justify ME module takes major encoding time, d can also be approximated as a function of λ and θ. Figure 5-2 shows the ME time ratio in the whole encoding time for 3 sequences: foreman (qcif), akiyo (qcif) and stefan (cif), under different search range. We choose these three video sequences because they are good representatives for content categories of complicated motion, slow motion and fast motion respectively. Note that such ratio is obtained based on exhaustive full motion search, and may vary if any fast search algorithm is used instead. Nevertheless, we assume full motion search throughout the chapter to guarantee we always find the optimal motion vector. It can be seen that ME takes more than 70% encoding time (complexity) regardless of video content, and the ratio grows fast with search range increase. Therefore, it is reasonable we approximate encoding time by ME time in our model.

We now begin to analyze the functional relationship between distortion D and ME complexity parameter λ and θ. Since we do not have any prior knowledge of the function, the method we use is to actually draw the relationship between the two through experiments and see if it can be fitted with certain known functional form. Because the

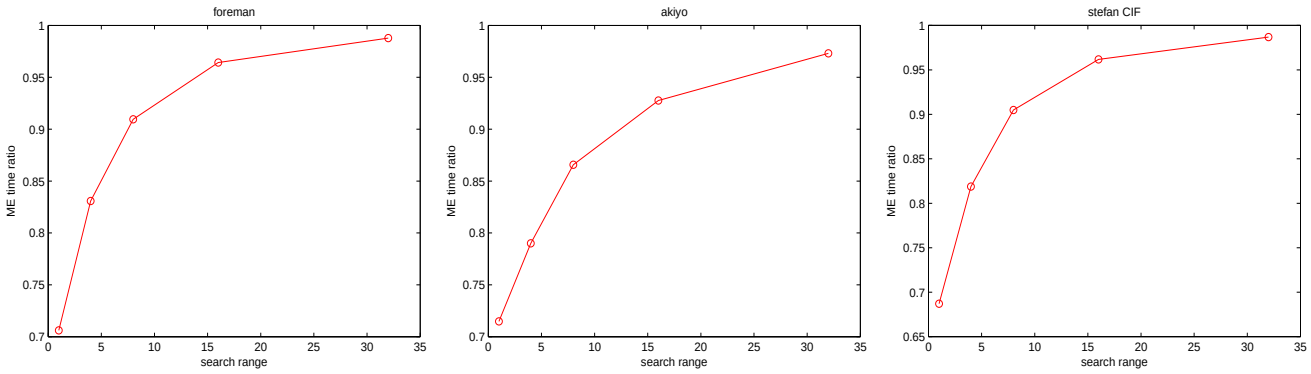


Figure 5-2. Motion estimation time ratio vs. search range. (a)Foreman (qcif). (b)Akiyo (qcif). (c)Stefan (cif).

inter coding mode has potential effect on the functional relationship between D and ME complexity parameter, we exclude its effect by specifying each inter mode in experiment except skip mode, as skip mode cannot stand alone in coding without support of other modes. However, we need to keep it in mind that in practical H.264/AVC encoding process, each MB can choose its own inter mode based on rate-distortion optimization (RDO), and the overall residue variance may not fit into any of the 7 functional forms of basic mode. This problem will be addressed later.

We implement the experiments in H.264/AVC JM16.0 software, and compute the average residue variance σ given 2-D motion search range λ and number of reference frame θ. Figure 5-3 shows the average residue variance vs. 2-D motion search range for 3 sequences: Foreman (qcif), Stefan (cif) and RaceHorses (WVGA). The inter mode is indicated by index 1 to 7 as in H.264/AVC configure. The 2-D search range in Figure 5-3 differs from that in Figure 5-2, but related by the following function:

$$\lambda = (2\lambda_0 + 1)^2 \tag{5–3}$$

where λ_0 is the search range value in Figure 5-2. the reason we map the parameter in this way is for the convenience of fitting the $d - R - D$ model. From Figure 5-3, we can identify the following facts: 1)The functional form between residue variance and 2-D

Table 5-1. Exponential Fitting Parameters of Residue Variance vs. 2-D Motion Search Range in Foreman Under Inter Mode 1,4,7

mode	a	b	c
1	161.8	0.1596	65.99
4	138	0.1668	42.77
7	116.6	0.1814	29.91

motion search range resembles exponential. 2)The functional forms of all 7 basic inter modes are very similar. 3)The functional forms of mode 2 (16×8) and 3 (8×16) are identical, and mode 5 (8×4) and 6 (4×8) are identical, with exception in Stefan, which is not identical but close.

Therefore, we can use exponential function to fit the residue variance vs. 2-D motion search range function. The exponential function for fitting is:

$$y = ae^{-bx} + c \tag{5–4}$$

with 3 fitting parameter $\{a, b, c\}$. We select the fitting results for Foreman under inter mode 1, 4, 7 and show them in Figure 5-4, with the corresponding fitting parameters in Table 5-1.

Likewise, we investigate the functional relationship between residue variance σ and reference frame number θ from experiments. The results of average residue variance σ vs. θ for Foreman (qcif) is shown in Figure 5-5, with fixed 2-D search range $\lambda_0 = 8$ ($\lambda = 289$). We observe that such a functional form can also be fitted by exponential in (5–4). The fitting results for inter mode 1,4,7 are given in Figure 5-6, and Table 5-2 lists the corresponding fitting parameters. In fact, 2-D motion search range λ multiply number of reference frame θ forms 3-D motion search cube of ME. But we have to separately evaluate the effect of λ and θ on the residue variance because the experimental results show that they are weighted very differently through exponential fitting parameters. Also, in H.264/AVC, λ and θ are two individual tunable ME parameters, rather than the 3-D ME cube. For purpose of clear illustration, we obtain the residue variance change by tuning

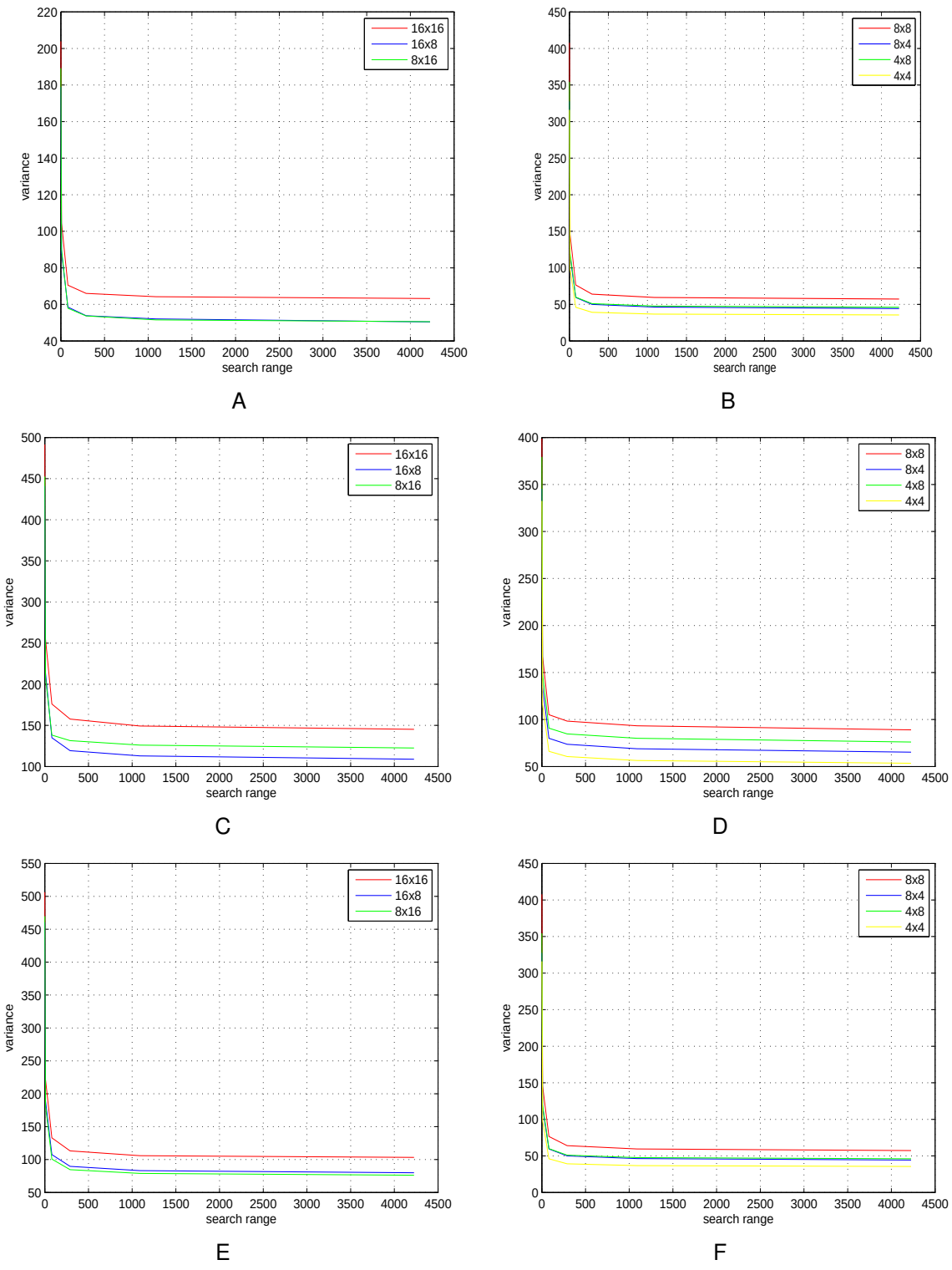


Figure 5-3. Residue variance vs. 2-D motion search range under 7 basic inter modes in H.264/AVC. (a)Foreman mode 1,2,3. (b)Foreman mode 4,5,6,7. (c)Stefan mode 1,2,3. (d)Stefan mode 4,5,6,7. (e)RaceHorses mode 1,2,3. (f)RaceHorses mode 4,5,6,7.

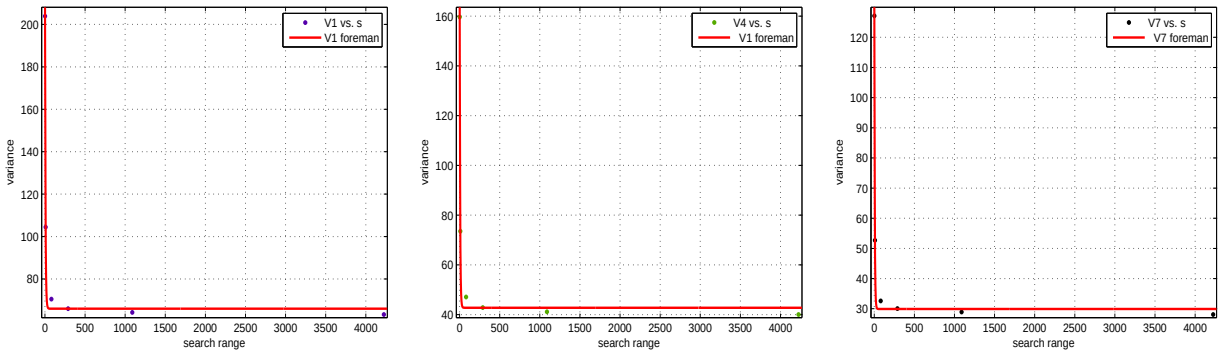


Figure 5-4. Exponential fitting of residue variance vs. 2-D motion search range in Foreman under inter mode 1,4,7. (a)Mode 1. (b)Mode 4. (c)Mode 7.

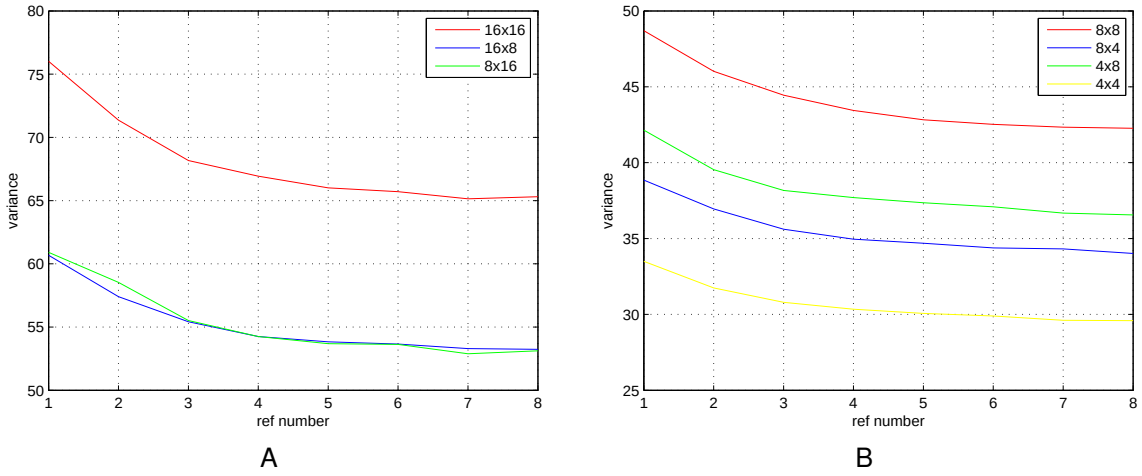
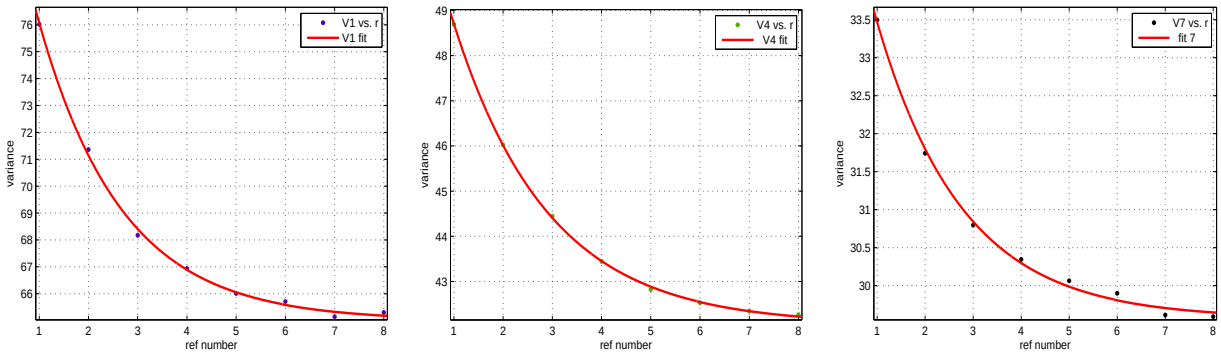


Figure 5-5. Residue variance vs. reference frame number under 7 basic inter modes in H.264/AVC for Foreman (qcif) (a)Mode 1,2,3. (b)Mode 4,5,6,7.

one parameter while fixing the other in the experiments. However, in real applications, we can tune the two parameters at the same time to achieve delay, rate and distortion requirement if the functional forms of both parameters are known.

Table 5-2. Exponential Fitting Parameters of Residue Variance vs. Reference Frame Number in Foreman Under Inter Mode 1,4,7

mode	a	b	c
1	19.96	0.5895	65
4	11.18	0.5177	42.04
7	6.823	0.5588	29.57



A B C

Figure 5-6. Exponential fitting of residue variance vs. reference frame number in Foreman under inter mode 1,4,7.(a)Mode 1. (b)Mode 4. (c)Mode 7.

Summarize the results above, we have the following functional form between residue variance and ME complexity parameters λ, θ,

$$\sigma = ae^{b_1\lambda + b_2\theta} + c \tag{5–5}$$

where a, b_1, b_2, c are fitting parameters. In [89], the authors derive distortion for i.i.d zero-mean Laplacian residual distribution under uniform quantizer as a function of quantization step size Q and Laplacian standard deviation ρ,

$$D(Q,\Lambda) = \frac{\Lambda Q e^{\gamma\Lambda Q}(2 + \Lambda Q - 2\gamma\Lambda Q) + 2 - 2e^{\Lambda Q}}{\Lambda^2(1 - e^{\Lambda Q})} \tag{5–6}$$

where γ is quantization offset, and

$$\Lambda = \frac{\sqrt{2}}{\sigma} \tag{5–7}$$

is a one-to-one mapping of σ used in the expression for simplicity. And Ref [90] derives entropy of transform coefficients with i.i.d zero-mean Laplacian distribution as

$$H(Q,\Lambda) = -P_0 \log_2 P_0 + (1 - P_0)(\frac{\Lambda Q \log_2 e}{1 - e^{-\Lambda Q}} - \log_2(1 - e^{-\Lambda Q}) - \Lambda Q\gamma log_2 e + 1) \tag{5–8}$$

where P_0 is the probability of zero after quantization

$$P_0 = 2\int_0^{Q(1-\gamma)} p(x)dx = 1 - e^{-\Lambda Q(1-\gamma)} \tag{5–9}$$

The entropy of quantized transform coefficients is actually entropy coding rate R in unit of bit per pixel (bpp), i.e.

$$R(Q,\Lambda) = H(Q,\Lambda) = -P_0 \log_2 P_0 + (1-P_0)(\frac{\Lambda Q \log_2 e}{1-e^{-\Lambda Q}} - \log_2(1-e^{-\Lambda Q}) - \Lambda Q\gamma log_2 e + 1) \tag{5–10}$$

When we plug (5–5) into (5–6) and (5–10), we derive coding distortion and rate as functions of quantization step size Q and ME complexity parameters λ, θ,

$$\begin{aligned} & D(Q,\Lambda) \\ & R(Q,\Lambda) \\ \Lambda \quad = \quad & \frac{\sqrt{2}}{\sqrt{ae^{b_1\lambda+b_2\theta}+c}} \end{aligned} \tag{5–11}$$

Meanwhile, the encoding time for one frame d can be written as

$$d(\lambda,\theta) = \frac{N\lambda\theta c_0}{\sqrt[3]{kP}} \tag{5–12}$$

where N is the number of MB in that frame,$\lambda\theta$ is the number of SAD operations utilized by ME per MB, c_0 is the number of clock cycle taken by each SAD operation for a given CPU. P is CPU power, and k is a constant in CPU power model called DVS in [91][92], which dynamically controls the energy consumption of microprocessor on portable devices. (5–12) means the encoding time is the total number of CPU clock cycles used in SAD operation divides the number of clock cycle per unit time.

Based on the d-R-D model above, we draw some figures to show the model estimate. Figure 5-10 gives a 3-D Pareto surface of source encoding time, rate and distortion. This is the achievable optimal distortion D as a function of encoding time d and rate R, which is consistent with our intuition that D decreases with more time

consuming coding (exhaustive ME search in IPPPP mode) and higher coding rate. To view the d-R-D model in more detail, we plot the D-d curve of Foreman under 8×8 inter mode in Figure 5-7, given various quantization parameters QP. According to H.264/AVC standard, QP and Q is a one-to-one mapping related as below:

$$Q = q_i \cdot 2^{\lfloor \frac{QP}{2} \rfloor} \tag{5–13}$$

Generally, it is concluded that D decreases with more exhaustive motion search under different quantization step size Q. However, for small Q, the descending curve becomes quite flat due to the small quantization distortion given fine granular of quantization step size, regardless of motion search. On the other hand, because the testing sequence we use is of quite small resolution (qcif), as well as none fast motion video content in Foreman, the optimal motion vector may be achieved at a certain search range value, and the enlargement of search range beyond that value will not contribute to distortion decrease. Hence the D curve becomes more flat after a certain encoding time in Figure 5-7. Likewise, we show the R-d curve given various quantization parameters QP of under the same testing condition for Foreman (qcif) in Figure 5-8. Because the more exhaustive motion search is more likely to find the optimal motion vector, residue variance decreases with longer encoding time d in IPPPP coding, leading to fewer bitrate given the same QP. On the other hand, given the same ME search range that achieves the same residue variance, bitrate will increase with finer quantization step size, i.e. smaller QP. Figure 5-9 illustrates the R-D performance under various encoding time for Foreman (qcif) under 8×8 inter mode. Assume the frame rate is 30 fps, and the maximum coding delay per frame is 33ms. The time use percentage represents the ratio of real encoding time to 33ms. For example, $d = 0\%$ indicates no ME is conducted, and $d = 100\%$ use indicates ME takes up the maximum time available (33ms). For a given d, D decreases with R, as in Shannon's source coding theory [93]. It is interesting to notice that in all pre-existing works, the source coding R-D model or

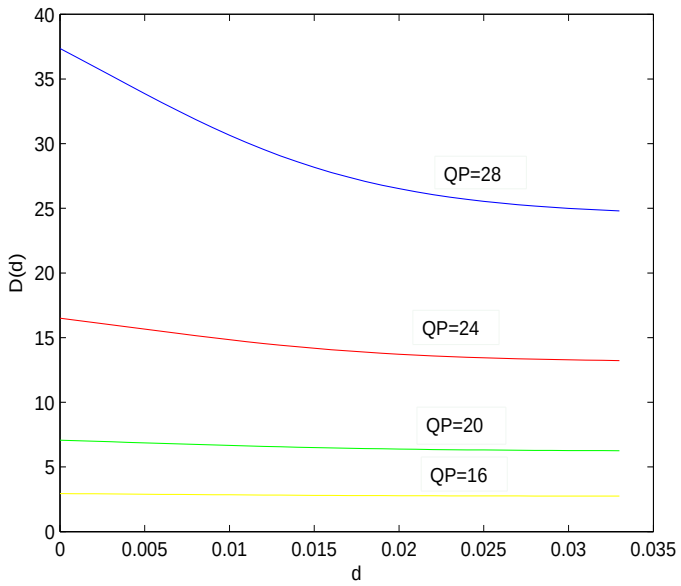


Figure 5-7. Distortion D vs. encoding time d given various quantization parameters QP in Foreman (qcif) under inter mode 4 (8×8).

the R-D performance only shows one of the R-D curves from a group of R-D curves with a fixed encoding time d as in Figure 5-9. The truth is for every achievable encoding time d, there is a corresponding Shannon source coding R-D curve. And any rate control based on a R-D model only provides a single R-D curve under a fixed but not specified d. In short, given $d(\lambda, \theta)$, the residue variance becomes constant in D and R in (5.1.2). Therefore, D and R becomes functions of quantization step size Q, and we can do rate control by tuning Q, which is the work in [89], and the methodology of some of the most important works in source coding rate control schemes as in [89][94][95][96][97] [98][99]

From the simulation results above, we have successfully modeled the source coding distortion, rate and delay, so that given a fixed encoding time, we can tune the encoding parameters of λ, θ and Q to get the minimum distortion under the rate constraint. To optimize the end-to-end performance of video communication system over wireless channels, we have to take transmission part into consideration, and analyze the relationship between transmission distortion and transmission delay given other system parameters.

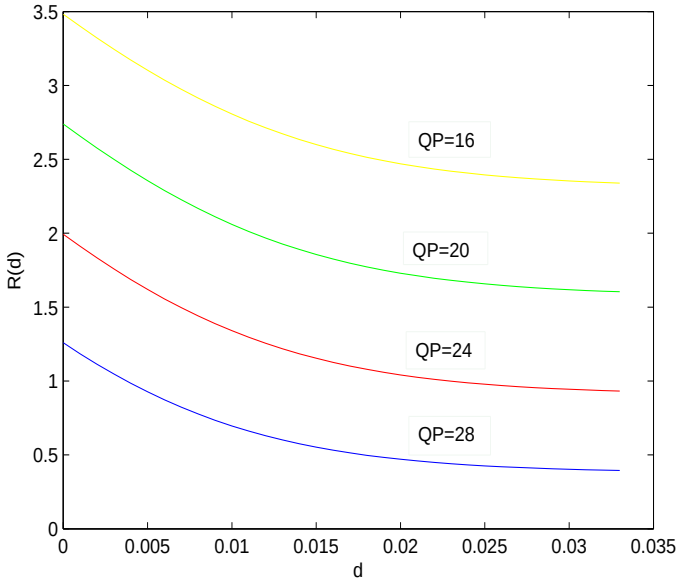


Figure 5-8. Rate R vs. encoding time d given various quantization parameters QP in Foreman (qcif) under inter mode 4 (8×8).

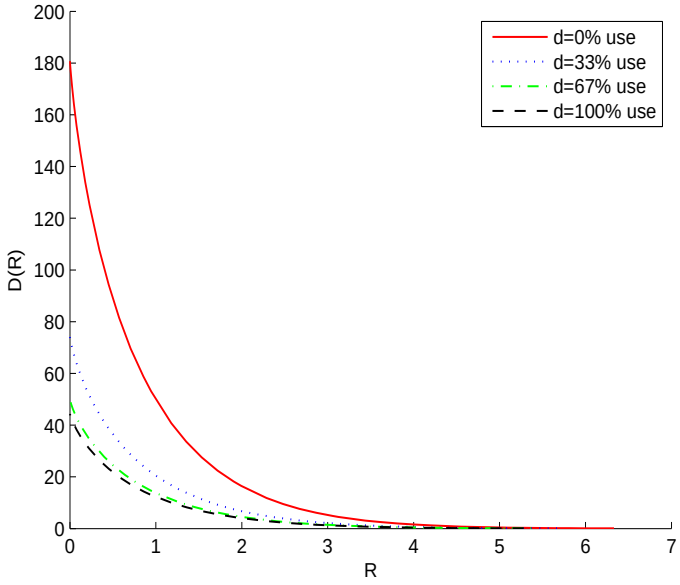


Figure 5-9. Distortion D vs. rate R given various encoding time d in Foreman (qcif) under inter mode 4 (8×8).

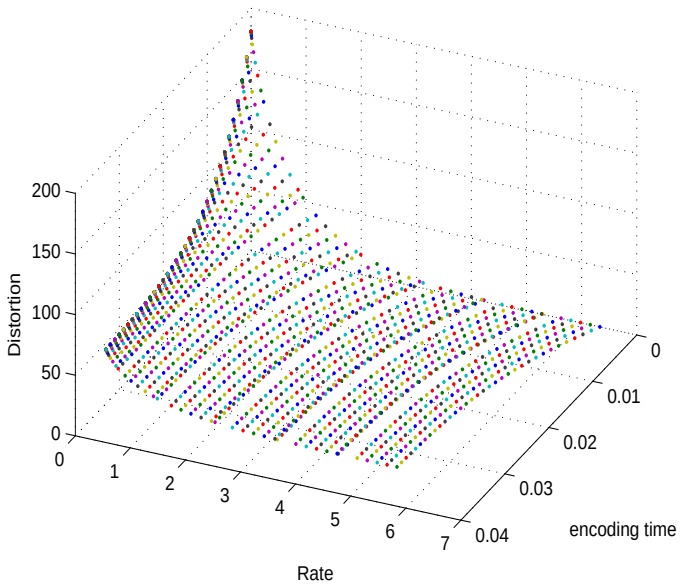


Figure 5-10. 3-D Pareto surface of Distortion D, rate R and encoding time d of d-R-D model.

5.1.3 Model Accuracy Verification

It is observed that though the exponential model fits well the functional form of the residue variance vs. ME complexity parameters for each of the 7 basic inter mode, we have to verify the model accuracy under real coding scenario in which each MB can choose from any inter coding mode including skip mode. The necessity to verify d-R-D model in real coding scenario are two folded. 1)For each frame, the percent of MB that chooses a certain inter mode is a random variable, therefore the residue variance will be a random combination of residue variance value from each inter mode. Since we do not have any knowledge of this random variable, we cannot predict the average residue variance. 2) Note that the value estimated in exponential model is the average residue variance of the entire video sequence, not every frame, hence we do not have the accurate residue variance of each inter mode that can be plugged into the combination for that frame, even if the percent of MB choosing each mode is not random. 3)Using skip mode, the independence of each inter mode does not hold, hence the true residue variance won't be a combination of 7 basic inter modes.

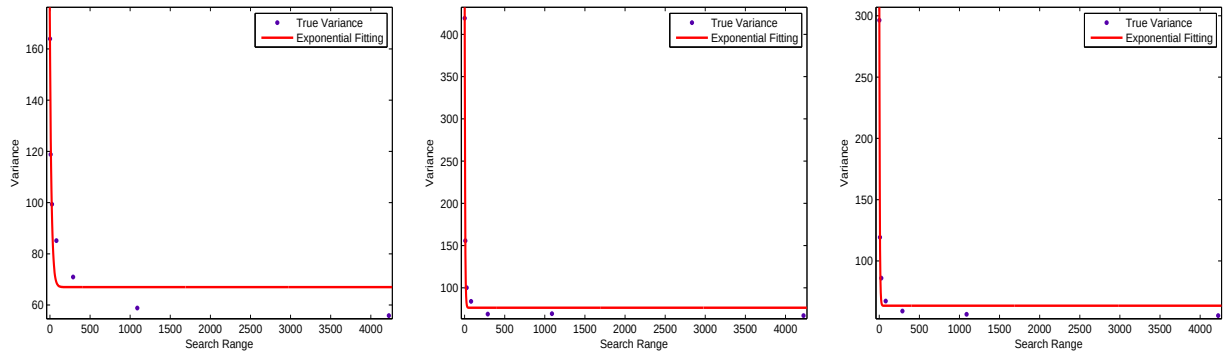


Figure 5-11. Exponential fitting of true residue variance vs. 2-D motion search range in IPPPP coding in H.264/AVC. (a)Foreman (qcif). (b)Stefan (cif). (c)RaceHorses (WVGA)

Again, we employ simulation results in H.264/AVC to investigate the relationship between true residue variance in IPPPP coding and ME complexity parameters. We test three sequences: Foreman (qcif, 100 frames, 30f/s), Stefan (cif, 90 frames, 30f/s), RaceHorses (WVGA, 100 frames, 30f/s), and set QP=28, $\theta = 4$. We vary the search range from 0-32, with 7 testing points $\lambda_0 = 0, 1, 2, 4, 8, 16, 32$, and record the residue of each coding P frame (exclude the first intra frame) and get the average residue variance. We observe the residue variance decrease with search range also looks like exponential form, so we fit it as in 5–4. Figure 5-11 presents the true residue variance vs. 2-D motion search range. It can be seen that though the functional form resembles exponential like every inter mode, the pure exponential fitting is not very successful. We notice that the exponential fitting is quite good at small search range, but the mismatch increases when $\lambda_0 > 16$. We also find that the mismatch difference is around $+10$ at points $\lambda_0 > 16$. Accordingly, we simply compensate the fitting on these points by -10, and we receive the corrected fitting results in Figure 5-12. It can be seen the corrected exponential fitting is much more accurate than the exponential fitting in Figure 5-11.

Similarly, to verify the model accuracy of residue variance vs. reference frame number, we conduct experiment in H.264/AVC to record true residue variance with θ. Unlike 2-D motion search range, the effect of MB inter mode selection on variance with θ

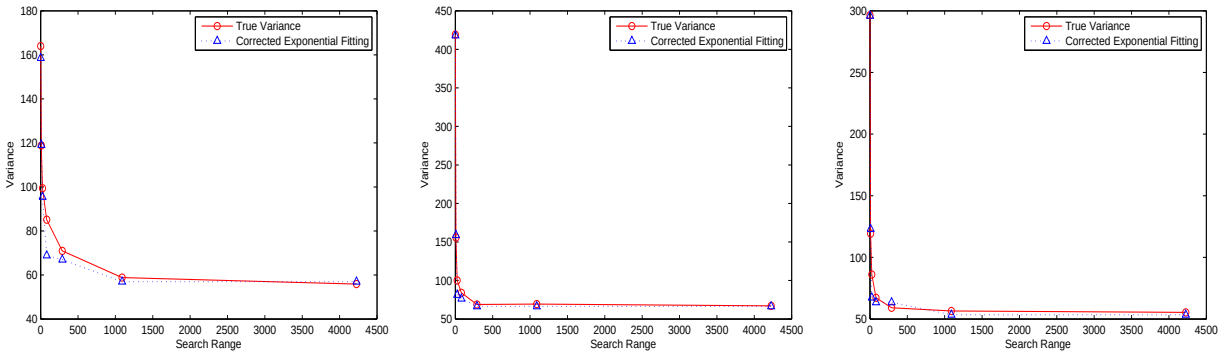


Figure 5-12. Corrected exponential fitting of true residue variance vs. 2-D motion search range in IPPPP coding in H.264/AVC. (a)Foreman (qcif). (b)Stefan (cif). (c)RaceHorses (WVGA)

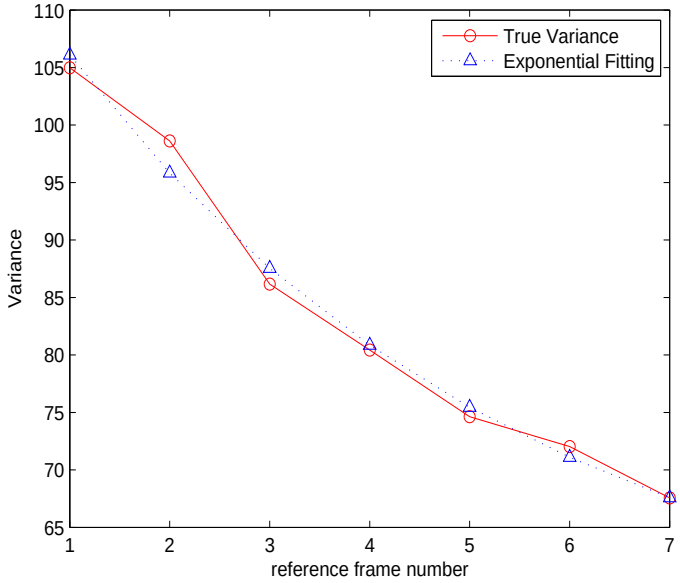


Figure 5-13. Exponential fitting of true residue variance vs. reference frame number in IPPPP coding in H.264/AVC for Foreman (qcif).

change is much less obvious, and the exponential fitting in (5–4) is able to give results of acceptable accuracy, as shown in Figure 5-13, where we set $QP = 28$, $\lambda_0 = 4$.

Having confirmed the exponential model of residue variance vs. ME complexity parameters is accurate for real coding scenario, we move on to confirm the model accuracy of d-R-D. For encoding time d, since we implement all experiments on H.264/AVC reference software JM16.0, which runs very slow in PC and cannot meet

the requirement of real time encoding application like 30 fps, it is not meaningful to compare the model estimate d with the true encoding time. So we will only verify the model estimate of coding distortion D and rate R. We compare the model estimate of D in (5–6) with coding distortion, i.e. distortion between reconstructed video and original video in the sense of mean square error (mse), and model estimate of R in (5–10) with coding rate in unit of bit per pixel.

Figure 5-14 to Figure 5-17 show the model estimate of D and R vs. motion search range λ, comparing with the true mse and bpp value in coding. Only one inter coding mode is enabled for each testing sequence, to be specific, 16×16 and 8×8 for Foreman (qcif) in Figure 5-14 and Figure 5-15, 16×16 for Stefan (cif) in Figure 5-16 and 16×8 for RaceHorses (WVGA) in Figure 5-17. Note here we use the true residue variance without exponential fitting to avoid potential discrepancy brought by the exponential model. Similarly, Figure 5-18 shows the model estimate of D and R vs. reference frame number for Foreman (qcif), with motion search range fixed at $\lambda_0 = 4$. We find model estimate of D and R differ significantly with the true coding values. This indicates the D and R models in [89] and [90] are not accurate enough. Consequently, we have to improve the model accuracy of D, R as functions of residue variance and quantization step size.

In [90], it addresses the D, R model mismatch problem. The assumed residal probability distribution e.g. Laplacian distribution in (5–6), may deviate significantly from the true residual histogram, especially when the number of samples are not sufficient. Therefore, it proposes to compensate the mismatch between the true residual histogram and assumed Laplacian distribution to obtain a better model estimate. The basic idea is to use the ratio of true distortion to the estimate in previous frame to correct the estimate of the current frame. This is based on a strict assumption that the changes of residual statistics and quantization step size between adjacent frames have almost the same effect on true distortion and model estimate [90], which is usually the case in practical video coding in IPPPP mode. Denote D_t^{k-1} true distortion of frame $k-1$,

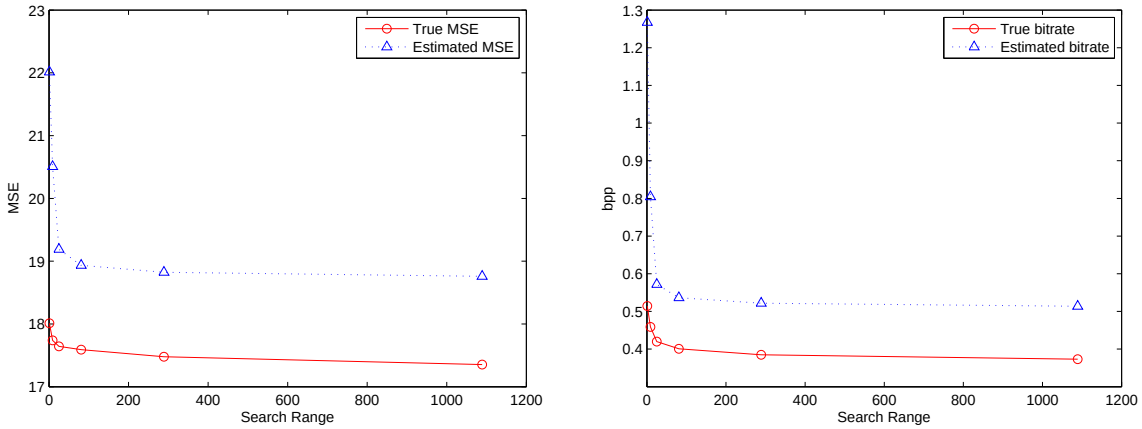


Figure 5-14. True coding D and R value vs. motion search range in comparison to model estimate of Foreman inter mode 1 (16×16), with reference frame number $\theta = 4$, $QP = 28$. (a)Distortion MSE. (b)Bitrate Bpp.

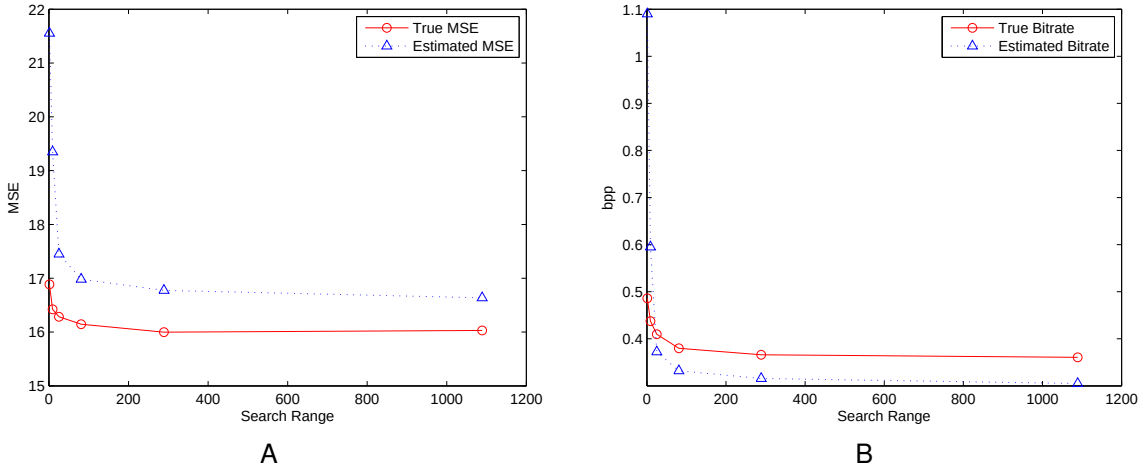
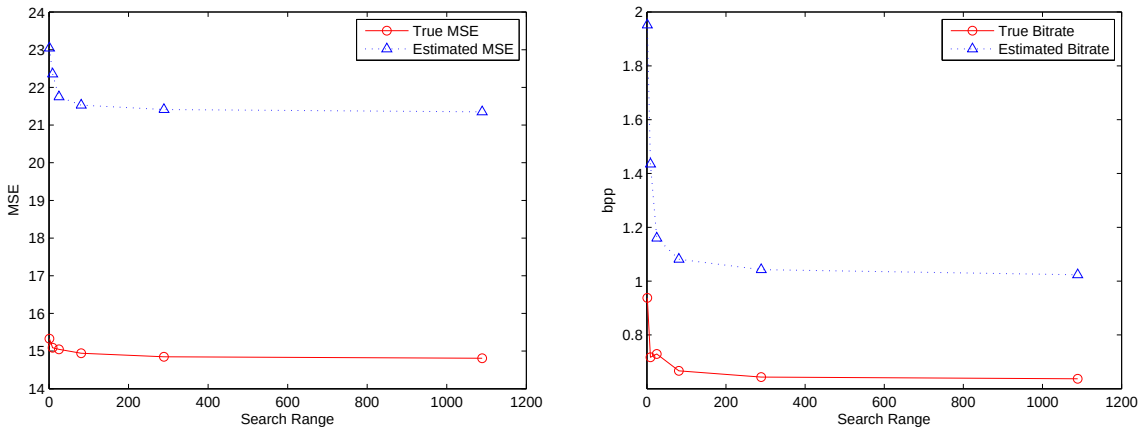


Figure 5-15. True coding D and R value vs. motion search range in comparison to model estimate of Foreman inter mode 4 ($8\times$), with reference frame number $\theta = 4$, $QP = 28$. (a)Distortion MSE. (b)Bitrate Bpp.

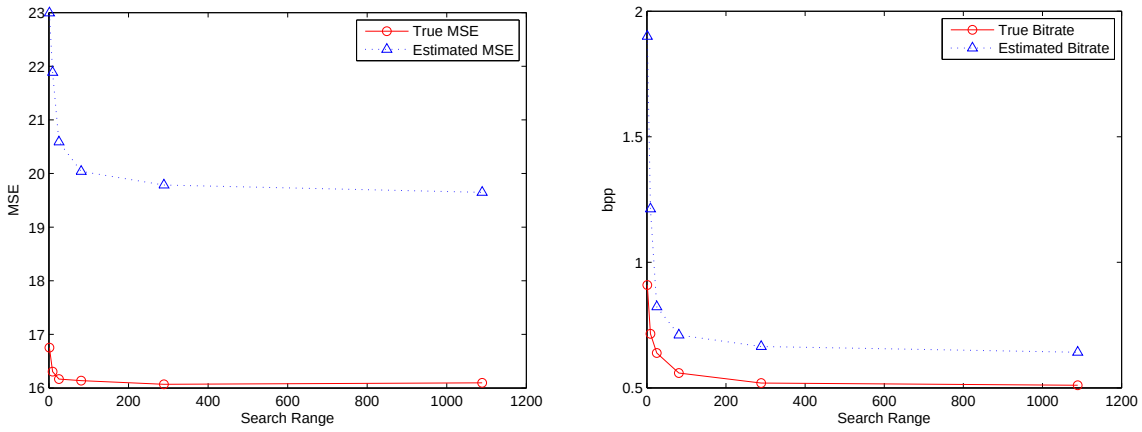
D_e^{k-1} the estimated distortion of frame $k-1$, where residue variance σ comes from exponential fitting model of (5–4). Hence, the distortion estimate of frame k can be further compensated as

$$\hat{D}_e^k = \frac{D_t^{k-1} \cdot D_e^k}{D_e^{k-1}} \tag{5–14}$$



A B

Figure 5-16. True coding D and R value vs. motion search range in comparison to model estimate of Stefan inter mode 1 (16×16), with reference frame number $\theta = 4$, $QP = 28$. (a)Distortion MSE. (b)Bitrate Bpp.



A B

Figure 5-17. True coding D and R value vs. motion search range in comparison to model estimate of RaceHorses inter mode 2 (16×8), with reference frame number $\theta = 4$, $QP = 28$. (a)Distortion MSE. (b)Bitrate Bpp.

Likewise, we compensate rate model estimate (5–10) in the same way as

$$\hat{R}_e^k = \frac{R_t^{k-1} \cdot R_e^k}{R_e^{k-1}} \tag{5–15}$$

The compensated model estimate is presented in Figure 5-21 to Figure 5-22, as long as the true coding mse, bpp and the model estimate before compensation for Foreman (qcif), Stefan (cif) and RaceHorses (WVGA). Different from Figure 5-14 to

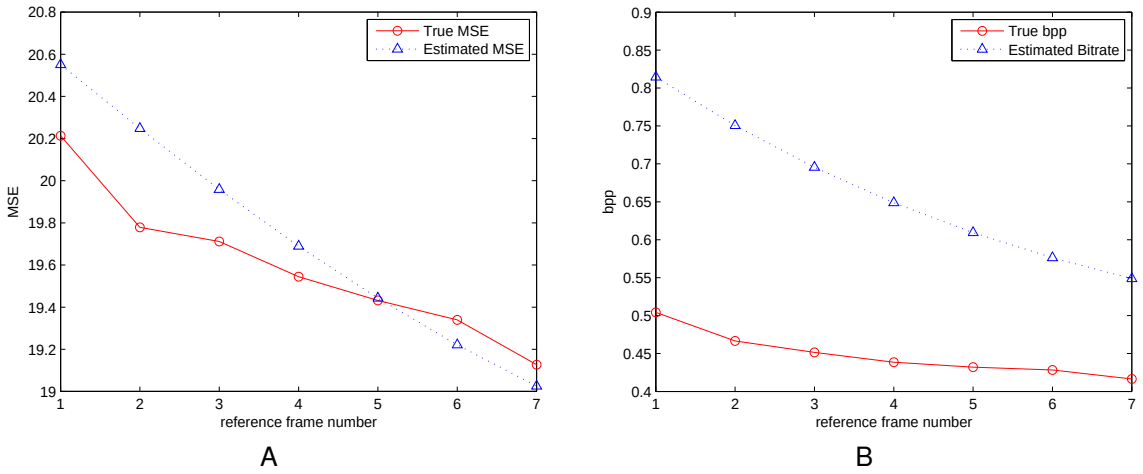
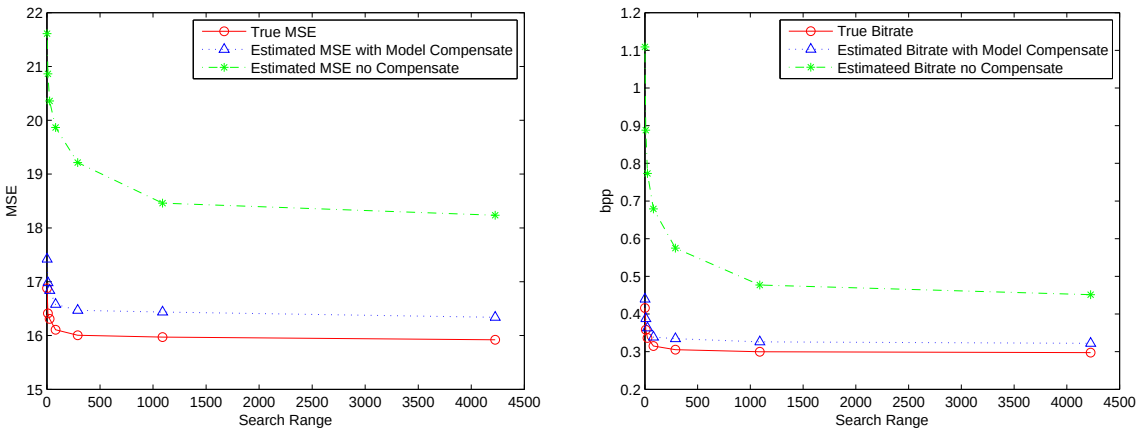


Figure 5-18. True coding D and R value vs. reference frame number in comparison to model estimate of RaceHorses inter mode 2 (16×8), with motion search range $\lambda_0 = 4$, $QP = 28$. (a)Distortion MSE. (b)Bitrate Bpp.

Figure 5-17, we employ the real coding in IPPPP mode where each MB is free to choose any of the 8 inter modes rather than all uniformly choose one mode. The true mse and bpp are coding values obtained under such coding scenario accordingly for fair comparison. All other coding conditions are the same as before. It is easily to see that there is huge improvement between true value and estimate with model compensation. Generally, the compensated model estimate can be good approximate to the true coding value, which verifies the proposed R-D model in (5–6) and (5–10) is accurate to be utilized with compensation process, so is the d-R-D model in (5.1.2).

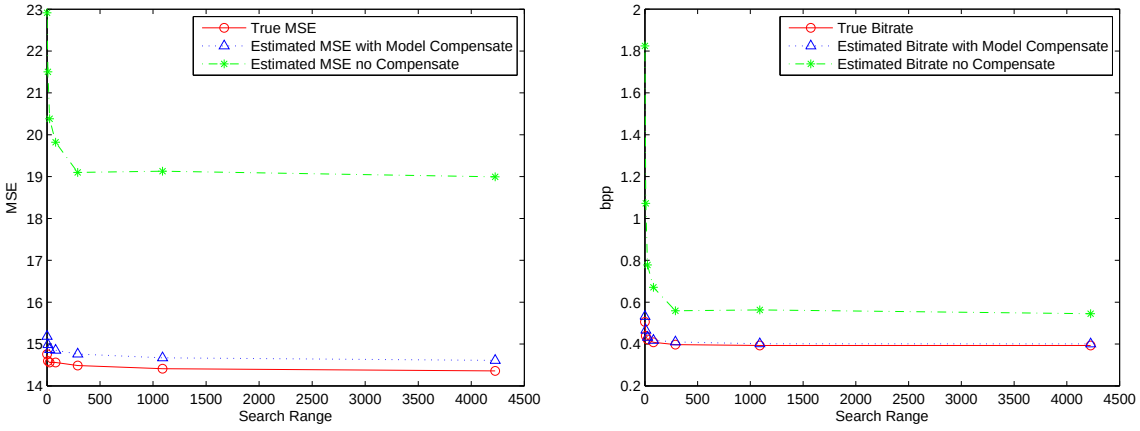
5.1.3.1 The effect of coding conditions on model accuracy

We have observed that without model compensation, the estimate of D and R is not accurate compared with true coding value. The results of true coding value we get in Figure 5-21 to Figure 5-22 is to enable all MB coding modes, sub-pel accuracy motion search and deblocking filter. This indicates each MB can choose from skip mode, 7 inter modes and intra modes as well. The effect of deblocking filter is usually ignored because it is very difficult to accurately modeled in R-D model [89][90][100]. Meanwhile, the R-D model in [89] does not take into account sub-pel accuracy motion search.



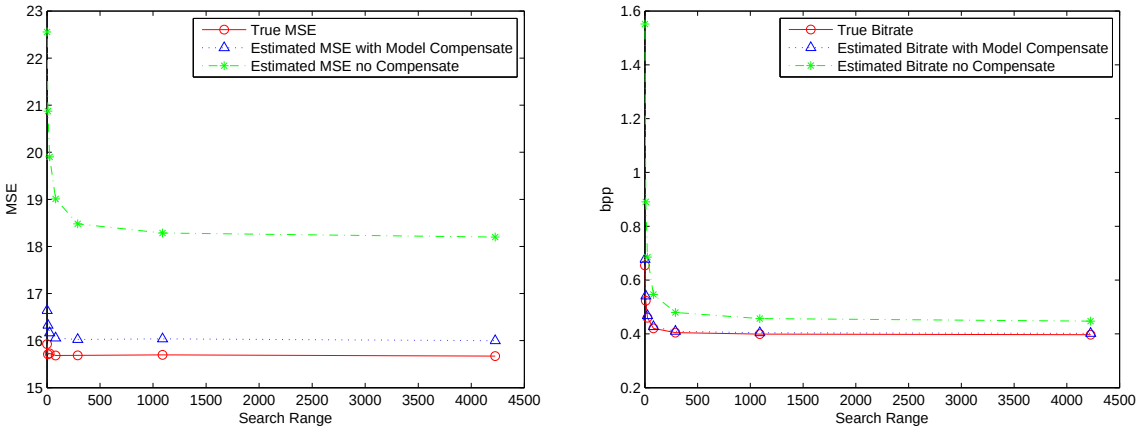
A B

Figure 5-19. Compensated model estimate D and R vs. motion search range of Foreman in real IPPPP coding compared with true distortion and rate, in contrast to uncompensated model estimate. (a)Distortion MSE. (b)Bitrate Bpp.



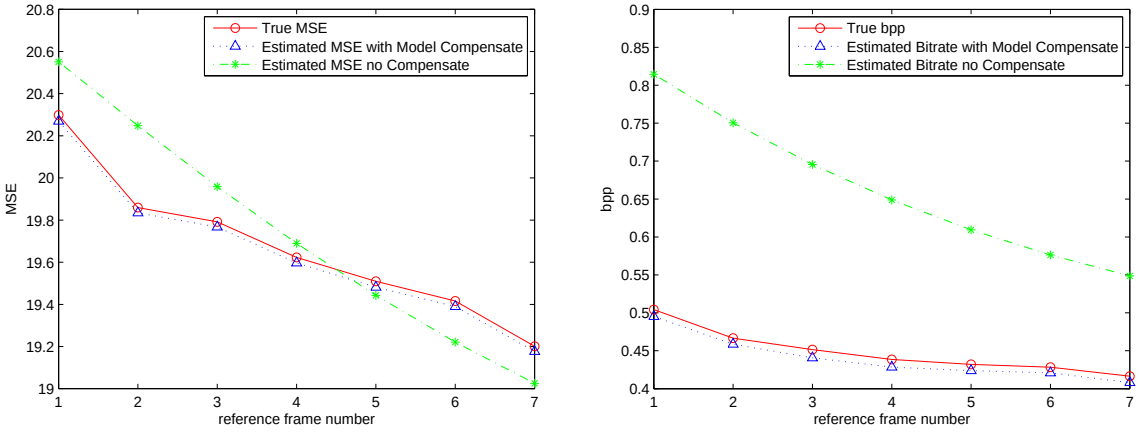
A B

Figure 5-20. Compensated model estimate D and R vs. motion search range of Stefan in real IPPPP coding compared with true distortion and rate, in contrast to uncompensated model estimate. (a)Distortion MSE. (b)Bitrate Bpp.



A B

Figure 5-21. Compensated model estimate D and R vs. motion search range of RaceHorses in real IPPPP coding compared with true distortion and rate, in contrast to uncompensated model estimate. (a)Distortion MSE. (b)Bitrate Bpp.



A B

Figure 5-22. Compensated model estimate D and R vs. reference frame number of RaceHorses in real IPPPP coding compared with true distortion and rate, in contrast to uncompensated model estimate. (a)Distortion MSE. (b)Bitrate Bpp.

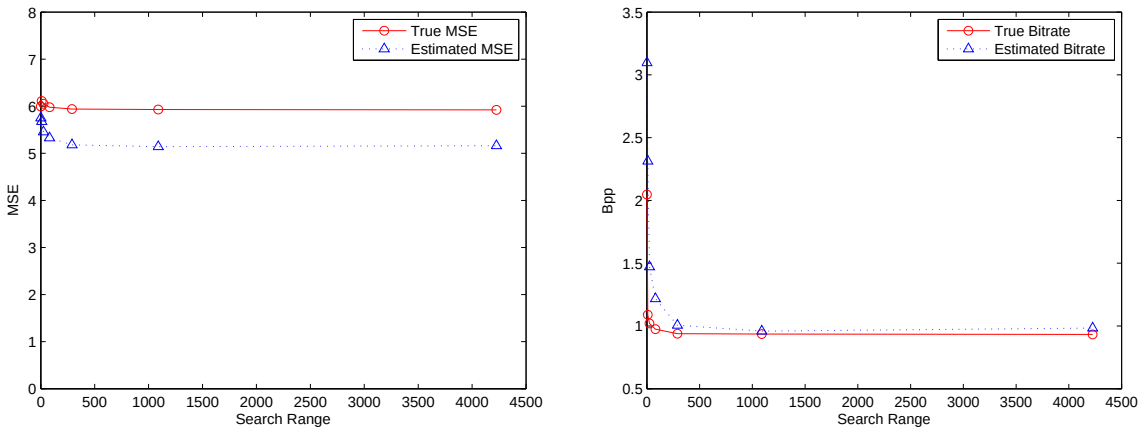


Figure 5-23. Model estimate D and R vs. motion search range of Foreman in real IPPPP coding, where MB intra mode, sub-pel motion search and deblocking filter is disabled, compared with true distortion and rate (a)Distortion MSE. (b)Bitrate Bpp.

Since we borrow the model structure in [89], subpel search should be disabled when verifying model accuracy. To this end, we disable MB intra mode in mode selection, sub-pel accuracy motion search and deblocking filter in H.264/AVC JM16.0 to compare the model estimate of D, R with the true coding value the second time. The results are shown in Figure 5-23. Obviously, the mismatch between the estimate and true value decreases, especially in bitrate estimate, which indicates these coding conditions have a major effect on the model accuracy.

5.1.3.2 The effect of coding parameters on model accuracy

In sections above, we verify the compensated model accuracy with only one coding parameter. For instance, in D and R model vs. motion search range, we fixed reference frame number $\theta = 4$, quantization parameter $QP = 28$. To detect the effect of these coding parameters on model accuracy, we conduct the following experiments.

First, we test residue variance vs. motion search range under different reference frame number ($\theta = 1, 4$) in Foreman (qcif), with $QP = 28$. The result is given in Figure 5-24. Note that the estimate is quite accurate regardless of the reference frame number. However,when search range is small ($\lambda_0 = 0$), the variance with $\theta = 4$ is much less than

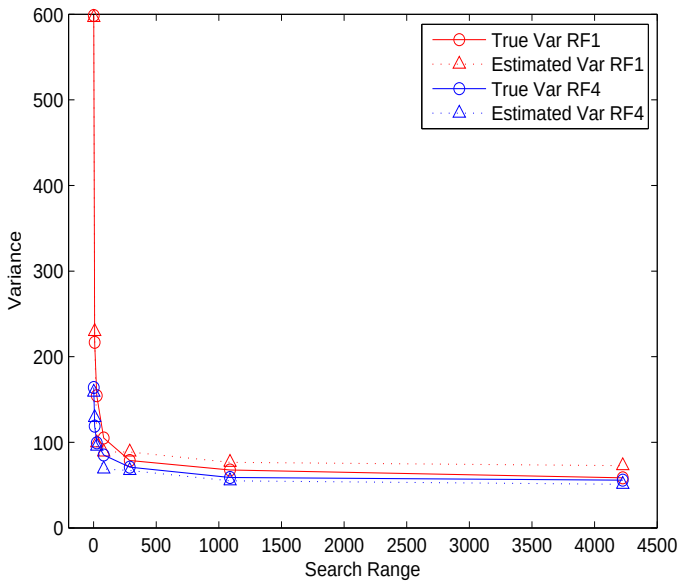


Figure 5-24. Exponential fitting of true residue variance vs. motion search range under different reference frame number for Foreman (qcif).

$\theta = 0$, but they are getting closer when search range enlarges. This means that when motion search range is reasonable, the number of reference does not play a significant role in the variance performance.

Next, we test residue variance vs. motion search range under different quantization parameter QP in Foreman (qcif), with $\theta = 4$. The result is given in Figure 5-25. Generally, the exponential fitting is accurate compared with the true variance regardless of QP. On the other hand, when QP gets large, the variance increases given search range. This means when λ is fixed, QP will play a key role in quantization distortion. Small QP corresponds to less distortion, and better decoded frame as reference, and hence smaller variance. Consequently, in the exponential model of residue variance, fiting parameters a, b, c should be re-estimated if QP changes.

How to estimate fitting parameters a, b, c in exponential model of residue variance (5–4) is a research problem itself. From the experiment results above, these parameters change with video content and quantization parameter QP. In rate control problem, the function of d-R-D model is to estimate the coding time, rate and distortion and tune the function parameters λ, θ and QP to meet the coding requirement. Hence,

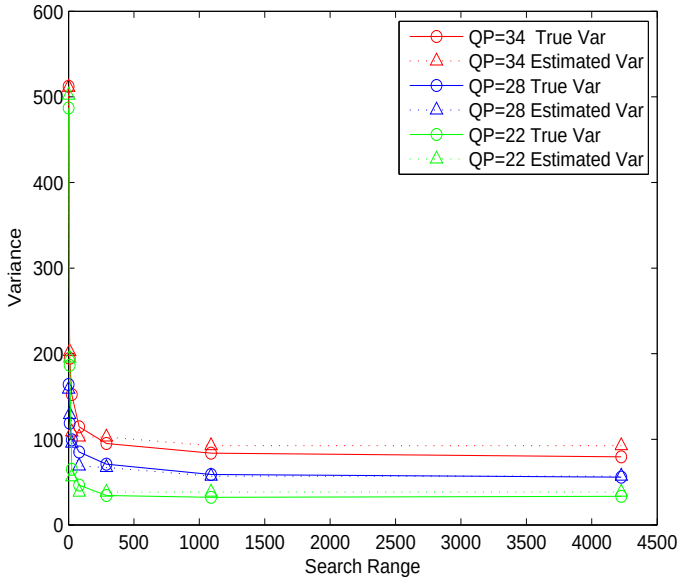


Figure 5-25. Exponential fitting of true residue variance vs. motion search range under different QP for Foreman (qcif).

the exponential model parameters that relates residue variance σ and λ, θ should be accurately estimated. In offline applications where real time rate control is not necessary, we can run a video sequence several times under different λ to get residue variance for each testing point if λ is the tuning parameter (if θ vice versa), and fit the residue variances into the exponential model to derive a, b, c. As illustrated above, a, b, c vary with QP, and re-estimate is required if QP or video content change. However, this method to estimate the fitting parameters is too slow and impractical for real time applications. To speed up the process in online mode, we first run a few frames in a sequence under different λ to get the residue variance for each testing point if λ is the tuning parameter (if θ vice versa) to get a rough estimate of the fitting parameters a, b, c. Generally, this is not accurate due to too few samples. Next, we estimate residue variance for each frame using the rough estimated parameters. Assuming that the residue statistics of adjacent frame does not change, we may use the previous frame statistics to compensate the current frame by the ratio of exponential model estimated residue variance to true residue variance after the previous frame has been coded. Denote σ_t^{k-1} the true residue variance of frame $k-1$, σ_e^{k-1} the estimated residue

variance of frame $k-1$, the estimated residue variance for frame k, σ_e^k can be further compensated as

$$\hat{\sigma}_e^k = \frac{\sigma_t^{k-1} \cdot \sigma_e^k}{\sigma_e^{k-1}} \tag{5–16}$$

A profound investigate into the estimate of model parameters for real time coding application is out of the scope of this work. The similar problems can be found in [88] and [90]. Please refer to these references for more details.

5.1.4 Rate-distortion Optimized Source Rate Control

With the d-R-D model we have developed in the previous two sections, we now apply our model to source rate control application. Note that there have been numerous works on source video coding rate control, and they all follow the same methodology: 1) a source distortion and source rate model is developed to accurately estimate the coding distortion and rate, and the distortion and rate are correlated through some coding parameters. In [96] and [95], it models distortion and rate as functions of quantization parameter, and some derived the model as function of video statistics and quantization parameter [99] [89] [101], where the video statistics in [89] is variance of source residual distribution, as is followed in our d-R-D model. The model in [94] derived rate and distortion as functions of video statistics and parameter ρ, the percentage of zero coefficient in transformed residue. In He's paper [88], it correlates rate, distortion and encoding power through some coding complexity parameters, hence distortion becomes a function of rate and coding complexity parameters. 2)rate-distortion optimization (RDO) method is applied to obtain the parameters that solve the rate control problem. It is worth pointing out that rate control and RDO are totally different concepts. The task of rate control is to compress the desired coding unit (can be sequence, GOP, frame and macroblock)within the bit-rate constraint. In theory, there are infinite solutions to this problem, and RDO is try to find one solution from the entire solution set that maximize the use of bit-rate to minimize the distortion.

The difference between the source rate control problem we have and those existing works is there are two constraints attached to the problem, one is the traditional rate constraint, the other is the encoding time constraint. The rate control problem can be formulated as

$$\begin{aligned} min \quad & D \\ s.t \quad & R \leq R_0, \\ & d \leq d_{max} \end{aligned} \tag{5–17}$$

where R_0 is bit-rate constraint, and d_{max} is encoding time constraint.

As we discussed in d-R-D model, source distortion, rate and encoding time are correlated through encoding complexity parameters, search range λ and reference frame number θ in IPPPP mode. Comparing Figure 5-3 and Figure 5-5, it can be seen that search range λ is a more effective parameter on residual variance change than reference frame number θ. To this end, we choose λ as the tuning parameter to model the relationship between source R-D and encoding time. Based on the d-R-D model in section 5.1.2, (5–17) can be further interpreted as below

$$\begin{aligned} min \quad & D(\lambda, QP) \\ s.t \quad & R(\lambda, QP) \leq R_0, \\ & d(\lambda) \leq d_{max} \end{aligned} \tag{5–18}$$

Note that we do not explicitly indicate the coding unit in (5–18) because it can be either sequence level or frame level rate control problem, which we will illustrate in greater details next. The key that differentiates the rate control problem on different coding units is the estimate from the model. If the model can accurately estimate the bit-rate and distortion of a sequence, RDO can be used to solve the sequence level rate

control problem. On the other hand, if the model is established to estimate the bit-rate and distortion of a frame, RDO will apply for frame level rate control.

5.1.4.1 Sequence level rate control problem

On a sequence level, the R-D model is supposed to accurately estimate bit-rate and distortion of a entire video sequence. However, depending on the granularity of parameter tuning capability, the sequence level rate control problem can be further divided to two cases: RDO with quantization parameter QP on sequence level and RDO with quantization parameter QP on frame level. Note that in (5–18), both λ and QP are tunable parameters, but on sequence level rate control problem, we only tune search range λ on a sequence level, i.e. set λ the same for all frames in a sequence because 1)all video encoder platforms compatible to H.264/AVC standard [102], such as JM [103] and x264 [104][105], does not support varying search range for each frame, but they do support quantization parameter to change on a frame level (JM) or even block level (x264) under traditional rate control scheme. 2)The R-D model as functions of λ only enable to accurately estimate the average bit-rate and distortion over the entire sequence. Therefore, the estimate of the residual variance σ for each frame in that sequence will stay unchanged, and RDO on the sequence level becomes RDO on the frame level since RDO for the sequence is the identical RDO for each frame adding up. To be more specific, each frame will choose the same λ and QP given sequence level R-D model, and hence it obtains the same λ for all frames in a sequence even if we allow λ to vary for each frame. Therefore, we will only discuss whether to vary QP on a frame basis in sequence level rate control problem.

1. Sequence level RDO with Sequence Level Parameter QP

In this case, the RDO can be formulated as

$$
\begin{aligned}
min \quad & D(\lambda_{seq}, QP_{seq}) \\
s.t \quad & R(\lambda_{seq}, QP_{seq}) \leq R_0, \\
& d^k(\lambda_{seq}) \leq d_{max}
\end{aligned}
\tag{5–19}
$$

where λ_{seq} and QP_{seq} indicate search range and quantization step size fixed for a entire sequence coding. Note encoding time constraint is always on a frame basis, for example, d^k is the encoding time of frame k, and d_{max} is the encoding time constraint of that frame. This is because frame input follows a regular period, i.e. a fixed frame rate like 30 frames/s, and the encoding time should be within this period to guarantee real time video coding.

To solve the rate-distortion optimization in (5–19), Lagrangian approach can be used. Following the theory in [106] [107], it transforms the RDO in 5–19 to another optimization problem

$$
\begin{aligned}
min \quad & J = D(\lambda_{seq}, QP_{seq}) + \Lambda R(\lambda_{seq}, QP_{seq}) \\
s.t \quad & \lambda_{seq} \leq \lambda_{max}
\end{aligned}
\tag{5–20}
$$

The constraint on λ_{seq} is just to ensure the encoding time of each frame does not exceed the frame interval. Considering λ_{seq} is discrete just as QP ($0 \leq QP \leq 51$), the constraint $\lambda_{seq} \leq \lambda_{max}$ explicitly points out what value λ_{seq} can take, (5–20) can be actually solved as a none-constraint optimization problem. Section 5.1.3 has successfully verified the model accuracy of bit-rate and distortion on a sequence level, i.e. the R-D as functions of λ and QP derived in section 5.1.2 can be plugged in (5–20). Λ in (5–20) is the preset Lagrangian multiplier, and can be determined either by bi-section search [90] [107] [108] or by modeling [103] [109] [89]. In this work, we propose an algorithm for sequence level source rate distortion optimization with bi-section search method to find the best Λ. This is because we

lack an accurate model of Lagrangian parameter and quantization parameter. In the previous modeling method [109] [89], they derived Lagrangian multiplier as a function of quantization parameter Q based on their specific R-D models. Note that how to derive such a model between Lagrangian multiplier and Q is still a research problem, and any inaccuracy of this model will cause inaccurate estimate of the optimal tuning parameter pair (λ, QP).

Algorithm 1: Sequence level rate-distortion optimized determination of search range λ_{seq} and quantization parameter QP.

Initialize $\Lambda_j = \Lambda_0$ using the method either in H.264/AVC JM software [103] or [107], $J_{min} = Inf$.

for $\Lambda_j = \Lambda_0, \Lambda_1, \cdots, \Lambda_N$ **do**
  for $\lambda_{seq} \leq \lambda_{max}$ **do**
    for $QP_{min} \leq QP_{seq} \leq QP_{max}$ **do**
      estimate D by (5–5), (5–7) and (5–6),
      estimate R by (5–5), (5–7) and (5–10),
      calculate J by (5–20),
      if $J < J_{min}$ **then**
        $J_{min} = J$,
        $QP_{seq}(\Lambda*) = QP_{seq}$,
        $\lambda_{seq}(\Lambda*) = \lambda_{seq}$
      end if
    end for
  end for
end for

obtain the best (λ_{seq}, QP_{seq}), i.e. ($\lambda_{seq}(\Lambda*)$, $QP_{seq}(\Lambda*)$.

We apply Algorithm 1 on test sequences Foreman and RaceHorses to find the optimal search range and quantization parameter on sequence level. The results are shown in Figure 5-26.

In Foreman sequence, we have set the maximum frame encoding time constraint and the other parameters in 5–12 so that allowable search range $0 \leq \lambda_{seq} \leq 32$. The rate-distortion optimization on sequence level determines the optimal parameter pair $(\lambda_{seq}*, QP_{seq}*) = (8, 24)$. It is a little bit surprising that the optimal search range is 8, not the largest 32. This is because the estimate cost by (5–20) at $\lambda_{seq} = 8$ is exactly the same as $\lambda_{seq} = 32$, both with $QP = 24$. Such a result has been verified in Figure 5-26A, where the picture quality with $\lambda_{seq} = 8$ and $\lambda_{seq} = 32$ is very competitive, where the average PSNR of luminance component for $\lambda_{seq} = 8$ is $38.9535dB$, while for $\lambda_{seq} = 32$ is $38.9723dB$. Though not exactly the same in actual coding, this estimate mismatch from the R-D model is considered acceptable. We also compare the picture quality when forcing $\lambda_{seq} = 4$, and RDO obtains optimal $QP = 24$ again. However, the average PSNR of Y is $38.9047dB$, close to λ_{seq} but still some room to improve. Similarly, we discover the optimal parameter pair for RaceHorses by (5–20) is $(\lambda_{seq}*, QP*) = (6, 31)$, when the search range constraint is $0 \leq \lambda_{seq} \leq 16$. Again, the RDO process does not choose the maximum search range 16, because the model estimate cost at $\lambda_{seq} = 6$ is exactly the same as $\lambda_{seq} = 16$, both at optimal $QP = 31$. The actual coding result in Figure 5-26B has demonstrated this result. The average PSNR of luminance component for $\lambda_{seq} = 6$ is $33.6115dB$, while for maximum allowable $\lambda_{seq} = 16$ is $33.6458dB$, which are very close. On the other hand, when forcing search range to be smaller than the optimal ($\lambda_{seq} = 2, 4$), the picture quality degrades quite a lot. Based on the obtained results, we can make the following conclusions:

1) The estimate cost derived from rate-distortion optimization is quite close to the actual coding result. This indicates the proposed d-R-D model has very good accuracy.

2) The RDO process can find the search range threshold over which the quality does not make much difference. For example, in Foreman sequence, the optimal search range derived from RDO is 8 instead of 32, which means to set search range greater than 8 will not contribute much in improving the coding efficiency, while any search range less than 8 leads to performance delinquency. Hence, the RDO derived optimal search range 8 can serve as a threshold. This is particularly useful in saving encoding time and computation power. As mentioned before, source coding complexity largely depends on motion estimation, while search range determines motion estimation complexity. If we knew the coding efficiency at a lower cost (smaller search range) is more or less the same as a higher cost (larger search range), we will not spend the valuable resources on extra ME, and thus save the power consumption and computation time.

3) The optimal parameters derived from RDO successfully reflect the feature of each individual video sequence. For instance, Foreman clip has low resolution and only contains flat motion in the tested first 10 frames, to enlarge search range will not have much impact on the coding efficiency as the best motion vector can be easily found in the adjacent block range. Therefore, the smaller than threshold $\lambda_{seq} = 4$ does not result in huge picture quality difference. On the contrary, RaceHorses has larger resolution and some fast motion in the clip, any search range smaller than the threshold will show potential performance degradation.

2. Sequence level RDO with Frame Level Parameter QP

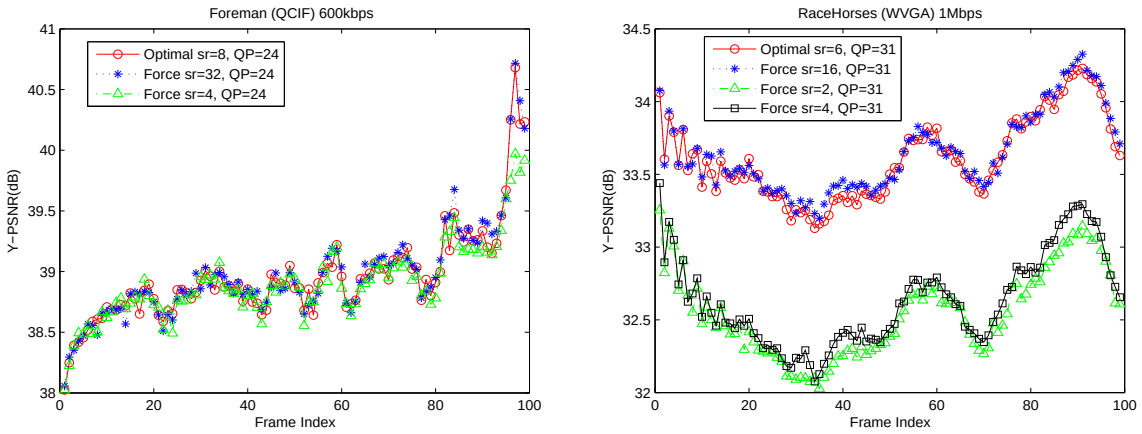


Figure 5-26. Picture quality with different parameter pairs of (λ, QP) on sequence level. (a)Foreman (QCIF) at $R_0 = 600kbps$; (b)RaceHorses (WVGA) at $R_0 = 1Mbps$.

As mentioned above, most of the widely used H.264/AVC coding platforms do not support search range varying on frame level, but they do support variable quantization parameter QP for each frame due to rate control. In conventional rate control, the control unit can be GOP, frame or block. RDO is utilized based on a R-D model as function of QP, so that by tuning respective QP of a control unit it can meet the target bit-rate while minimize the coding distortion. To this end, we can apply RDO based on proposed d-R-D model on sequence level rate control, but with frame level QP. Since each frame shares the same search range denoted as λ_{seq} while varying QP, the rate control can be carried out in the way identical to the conventional rate control scheme with R-D model in [89] when fixing λ_{seq}. In this case, the RDO can be formulated as

$$\begin{aligned} min \quad & \sum_{1}^{N} D^k(\lambda_{seq}, QP^k) \\ s.t \quad & \sum_{1}^{N} R^k(\lambda_{seq}, QP^k) \leq R_0 \\ & d^k(\lambda_{seq}) \leq d_{max} \end{aligned} \tag{5–21}$$

where N is number of frame in the sequence, search range is fixed for an entire sequence coding λ_{seq}, and QP^k indicates quantization step size varies for each frame in the sequence.

The Lagrangian method transforms RDO to another optimization problem

$$\begin{aligned} min \quad & J = \sum_{k=1}^{N} D^k(\lambda_{seq}, QP^k) + \Lambda \sum_{k=1}^{N} R^k(\lambda_{seq}, QP^k) = \sum_{k=1}^{N} (D^k + \Lambda R^k) \\ s.t \quad & \lambda_{seq} \leq \lambda_{max} \end{aligned} \tag{5–22}$$

Note that to minimize the sum of D and R of all frames is identical to minimize individual frame. The algorithm to solve (5–22) can be summarized below:

Algorithm 2: Sequence level rate-distortion optimized determination of search range λ_{seq} and frame level quantization parameter QP.

Initialize $\Lambda_j = \Lambda_0$ using the method either in H.264/AVC JM software [103] or [107], $J_{min} = Inf$.

for $\lambda_{seq} \leq \lambda_{max}$ **do**
  for $\Lambda_j = \Lambda_0, \Lambda_1, \cdots, \Lambda_N$ **do**
    for $k = 1, \cdots, N$ **do**
      set $J^k_{min} = inf$
      for $QP_{min} \leq QP^k_{seq} \leq QP_{max}$ **do**
        estimate D^k by (5–5), (5–7) and (5–6),
        estimate R^k by (5–5), (5–7) and (5–10),
        calculate $J^k = D^k + \Lambda_j R^k$,
        if $J^k < J^k_{min}$ **then**
          $J^k_{min} = J^k$,
          $QP^k(\Lambda*) = QP^k$,
        end if
      end for
    end for

end for

if $\sum_{k=1}^{N} J^k < J_{min}$ **then**

$J_{min} = \sum_{k=1}^{N} J^k$

$\lambda *_{seq} = \lambda_{seq}$

end if

end for

We observed from the algorithm above that when λ_{seq} is fixed, the RDO to find the optimal QP^k for each frame is exactly the same with the conventional RDO rate control scheme. It differs from conventional rate control that search range can vary on a sequence level within the time constraint. The concept of varying sequence level λ_{seq} can be extended to other R-D model, in which R, D are not necessarily function of search range, but still functions of QP, plus search range has impact on encoding time (most likely encoding time depends on coding complexity parameters like search range, though they are implicit in R-D model functions). However, because R-D model does not include λ, it cannot analyze how λ will affect the coding performance ahead of time. To find the optimal λ, we have to do actual encoding with rate control on for all possible λ, and then the optimal λ and QP^k corresponds to the coding process with the best performance.

Notice that in (5–22), it requires R and D functions on a frame level, while we only verified the d-R-D model in section 5.1.3 on sequence level. Therefore, we do not show experiment results on sequence level RDO with frame level QP. This will be one of our future works. Instead, we show the encoding results of different search range with rate control on for the first 100 frames in Foreman in Figure 5-27. Assume the maximum encoding time allows search range up to 16. We set search range at different values ($\lambda = 0, 4, 8, 16$), the first intra frame $QP = 27$, and apply original JM rate control [110] to the test sequence. The average PSNR for luminance component of the sequence under different search range is also marked

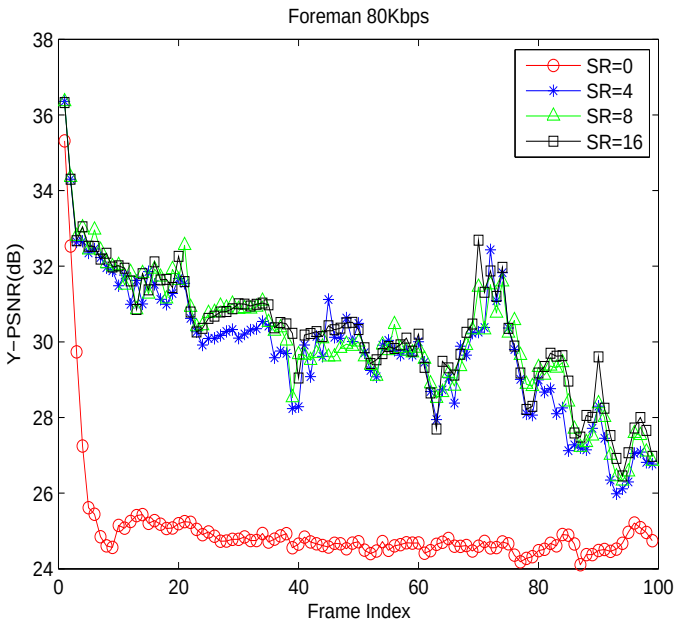


Figure 5-27. Picture quality of Foreman with rate control at 80Kbps in H.264/AVC over different search range (SR).

in the figure. Clearly, search range $\lambda = 16$ gives the best coding performance given original JM rate control, as we expect due to the finest residual error.

5.1.4.2 Frame level rate control problem

Though the proposed d-R-D model verified in the above sections is for a sequence, it can be extended to a frame. In this case, the fitting parameters of a, b, c plugging in to (5–6) and (5–10) has to be estimated on a frame level, i.e. these model parameters need to be adaptively updated for each coding frame, or a short period of time if a video has slowly-changing scene, in order to guarantee the model accuracy. How to estimate these parameters for each frame in a timely and accurate manner is a research topic itself, and is beyond the focus of this work. Suppose the rate distortion optimization for k-th frame, it can be formulated as

$$\begin{aligned} min \quad & D^k(\lambda^k, QP^k) \\ s.t \quad & R^k(\lambda^k, QP^k) \leq R_0^k \\ & d^k(\lambda^k) \leq d_{max} \end{aligned} \tag{5–23}$$

To solve (5–23), both Lagrangian approach and dynamic programming approach [111] are applicable. In terms of complexity, the Lagrangian approach is preferable since it can be run in linear time, whereas the complexity of dynamic programming can grow exponentially with the number of coding units considered [90] [111]. Because Lagrangian approach can be run independently in each coding unit, we can apply Lagrangian for k-th frame as

$$\begin{aligned} min \quad & J = D^k(\lambda^k, QP^k) + \Lambda R^k(\lambda^k, QP^k) \\ s.t \quad & \lambda^k \leq \lambda_{max} \end{aligned} \tag{5–24}$$

where D^k and R^k is distortion and rate model for k-th frame in a coding sequence, and the optimal variable pair $(\lambda^k, QP^k)*$ is also determined on a frame basis. The algorithm to solve (5–24) can be summarized below:

Algorithm 3: Frame level rate-distortion optimized determination of search range λ and quantization parameter QP for the k-th frame.

set $J_{min} = inf$
for $\Lambda_j = \Lambda_0, \Lambda_1, \cdots, \Lambda_N$ **do**
  for $\lambda^k \leq \lambda_{max}$ **do**
    for $QP_{min} \leq QP^k \leq QP_{max}$ **do**
      estimate D^k by (5–5), (5–7) and (5–6),
      estimate R^k by (5–5), (5–7) and (5–10),
      calculate $J = D^k + \Lambda_j R^k$,

if $J < J_{min}$ **then**

$J_{min} = J$,

$QP^k(\Lambda*) = QP^k$,

$\lambda^k(\Lambda*) = \lambda^k$,

end if

end for

end for

end for

The application of this frame level rate control scheme requires accurate frame rate and distortion model, i.e. model parameters a, b, c to be updated for each frame, which has not been achieved in the previous section. Furthermore, to adjust search range on a frame basis is to adjust the encoding time for a particular frame. Given the end-to-end delay of video transmission system, to adjust the encoding time of a frame will change the applicable transmission delay bound of the same frame, which affects the transmission distortion in an error-prone channel. Hence, it makes more sense to optimize the search range on frame level when the target is to minimize the end-to-end distortion in a video transmission system, rather than source coding alone. While in this work, we do not formulate functional relationship between frame transmission delay and frame transmission distortion in an erroneous channel such as wireless fading channel. Consequently, we do not provide any experiment result of frame level RDO with adaptable parameter pair (λ^k, QP^k). However, to justify that encoding time will affect end-to-end distortion (performance) when end-to-end delay is fixed, we will simulate a point-to-point video communication system, with search range varying on sequence level in following section.

5.1.5 Analysis of End-to-end Delay and Performance

In real time video communication system, it has a tolerable end-to-end delay. As described in section 5.1.1, the maximum end-to-end delay that each frame experiences

is a constant C. Follow the assumption that channel propagation delay is negligible, decoding buffer delay equals to encoder buffer delay as well as instant decoding, the tunable time in end-to-end delay model is simply encoding time ΔT_e and encoding buffer ΔT_{eb}, i.e.

$$\Delta T_e + \Delta T_{eb} = T \tag{5–25}$$

The proposed d-R-D source coding model in section 5.1.2 enables to vary encoding time ΔT_e by tuning ME complexity parameters in IPPPP coding of H.264/AVC, so as to change encoding buffer delay. Note source distortion and bitrate are functions of ΔT_e, while packet transmission performance packet error probability (PEP) is closely related to encoding buffer delay. This statement is particularly true in time-variant wireless communication channel. Generally, the average PEP has three sources: 1)packet drop error if the buffer is overflow. This will happen if the rate of buffer output cannot catch up with buffer input. 2)delay bound violation if the time a packet stays in buffer exceeds the maximum delay bound ΔT_{eb}. The time a packet stays in buffer is from the moment the packet is fed into the buffer to the moment it is pushed out into the channel for transmission. 3)decoding error provability if a packet is incorrectly decoded at the receiver. In real communication system, the bit error probability is always not zero due to channel noise. It is easy to see given a constant buffer length, there is a tradeoff between decoding error and the other two types of error. If we increase the service rate, the buffer will be cleared more quickly, resulting in a smaller packet drop probability and delay bound violation probability. However, the decoding error will increase since more bits are transmitted through the channel. So the optimal channel rate control policy is studied in [84] to balance the three types of error to minimize the total packet error probability. It analyzed each type of error mathematically to give the expression of the total PEP. A simpler case would be to set appropriate virtual buffer size so that the delay bound violation is negligible and omitted in the analysis of total PEP. If we assume the buffer input rate is a constant μ, the virtual buffer size should set to $M = \mu \times \Delta T_{eb}$

to avoid delay bound violation. It can be see that to change encoding buffer delay is equivalent to change virtual buffer size M. Follow the wireless fading channel model and packet drop probability analysis in Ref [77], we derived the relationship between packet drop probability and encoding buffer delay ΔT_{eb} in Appendix.

With d-R-D source coding model, the quality-of-service guarantee of end-to-end delay can be formulated as

$$\begin{aligned} min \quad & D_{ETE} = D_s(\Delta T_e, QP) + D_t(\Delta T_{eb}, PEP, r_s) \\ s.t \quad & \frac{R_s(\Delta T_e, QP)}{r_s} \leq R_0 \\ & T_e + T_{eb} = T \end{aligned} \tag{5–26}$$

The end to end distortion D_{ETE} is the addition of source distortion D_s and transmission distortion D_t, as proved in [112]. From d-R-D source coding model, D_s can be derived as function of encoding time ΔT_e and quantization parameter QP; transmission distortion is related to encoding buffer time ΔT_{eb}, PEP and channel coding rate r_s. While with brief analysis in Appendix, PEP is cross related to ΔT_{eb}. To solve the optimization problem in (5–26), we need to derive the model of transmission distortion $D_t(\Delta T_{eb}, PEP, r_s)$. Although [84] made a great effort to provide mathematical expression of PEP with ΔT_{eb}, it is not elegant to be employed in transmission distortion model due to two problems:

1. It assumes constant packet arrival rate or Poisson process with rate parameter μ. This indicates bitrate should be equally assigned among each frame, and can be realized only with source rate control. However, the source rate allocation is dependent on the overall system requirement of (5–26), which relies on encoding buffer input and output process, and in turn depends on packet arrival rate. This dilemma makes it impossible to achieve the rate control by one-pass encoding.

2. Of the three types of PEP derivation in [84], the delay bound violation probability and decoding error probability are given in form of an upper bound, rather than

a single value. Among them, the decoding error probability derivation accords to Shannon's channel coding theory utilizing random coding and decoding. Even this part is replaced by the convolutional coding and decoding, as derived in [90], it still appears as a theoretic bound. This makes a very rough PEP in transmission distortion $D_t(\Delta T_{eb}, PEP, r_s)$.

Therefore, to model transmission distortion as function of ΔT_{eb}, PEP and r_s is, if not impossible, a formidable task. In this work, we will analyze the transmission distortion through simulation. The simulated system diagram is shown in the Figure 5-1. The input is an uncompressed video sequence and the output is the decoded video sequence. The input video will be encoded frame by frame in a video encoder, H.264/AVC JM16.0 to be specific. The encoded frame is then packetized and fed into an encoder buffer in a queue, waiting to be transmitted. We use two wireless channel models, 1)a discrete-time additive white Gaussian noise channel (AWGN) and 2) a discrete-time, time-variant block-fading channel with additive white Gaussian noise which is often used to model slow-varying flat fading channels [77]. The received packets will be put into a decoder buffer. Once all packets from one frame have been collected into the decoding buffer, it will be decoded at the decoder. Given constant end-to-end delay T, and the error concealment scheme at decoder, tuning encoding time T_e changes encoder buffer delay T_{eb}. We will demonstrate how this change will affect the system performance in this simulation.

Suppose the input frame rate is 30 fps. To ensure real time communication, the maximum encoding time for each frame is no more than a frame duration in unit of ms, i.e.

$$0 < T_e < 33 \tag{5–27}$$

Hence, the encoder buffer delay should be $T_{eb} \leq T - T_e$ to meet end-to-end delay requirement. Intuitively, if T_e is small, the frame packets are allowed to stay longer in encoding buffer, reducing the chance to be dropped because of the delay bound

violation. Meanwhile, small T_e means very limited motion search range or fewer reference frame, resulting in increased source distortion based on results of source coding d-R-D model in section 5.1.2.

The detailed simulation settings are listed here. We force the encoding time of each frame to be identical and no more than 33ms, and packetized to several packets, with each packet the fixed length of 1000 bits. If the remaining bits is less than 1000 in the last packet, padding to 1000. Then, all packets from this frame will be poured into the encoding buffer to form a packet queue. To focus on how T_{eb} change will affect system performance, we neglect the packet drop due to buffer overflow by assuming an infinite length of encoding buffer. Thus, it restricts the packet drop situation to delay bound violation alone. Though this is impractical in real applications, it avoids the scenario where packets arrival rate overwhelms the channel output rate, resulting in too many dropped packets not from the change of T_{eb}, and still makes a valid simulation for analysis purpose.

Denote $\mathbf{x} = (x_1, x_2, \cdots, x_N)$ as channel inputs, and $\mathbf{y} = (y_1, y_2, \cdots, y_N)$ channel outputs. For discrete-time AWGN channel model,

$$y_m = \sqrt{g}x_m + n_m, m = 1, 2, \cdots N \tag{5–28}$$

and for discrete-time block fading channel,

$$y_m = \sqrt{g_n}x_m + n_m, m = 1, 2 \cdots N \tag{5–29}$$

where n_m is additive white Gaussian noise with variance $N_0/2$, and g_n is a memoryless channel gain process, i.e. $g_n(n = 1, 2, \cdots)$ are i.i.d with state space $\mathfrak{G} \subseteq \mathfrak{R}$ with probability density function of $\sqrt{g_n}$ a Rayleigh distribution

$$p(g) = \frac{g}{\sigma^2} e^{-\frac{g^2}{2\sigma^2}} \tag{5–30}$$

Based on Shannon's channel capacity [113], the channel capacity at gain $\sqrt{g_n}$ is

$$C(g_n) = Wlog_2(1 + \frac{P_{av}g_n}{N_0 W}) \tag{5–31}$$

where W is channel bandwidth, P_{av} is transmitting power. We apply a simple linear channel rate control policy $\mathcal{R}$ introduced in [84],

$$\mathcal{R}(g_n) = \rho \times C(g_n)\Delta_t / Lpackets \tag{5–32}$$

where Δ_t is time slot length, L is packet length. The simulation results show that the packet error probability is minimized at $\rho = 0.6$, and we will set $\rho = 0.6$ to balance the tradeoff between packet drop error and decoding error. Due to the channel noise, we choose convolutional code of rate $1/2$ for error correction, and Viterbi hard decoding at the receiver. Note that we do actual channel coding and decoding for each transmitted packet, and track the statistic of erroneous decoding error, rather than take the decoding error bound derived from Shannon's channel coding theory as in [84][77][90] to mitigate the estimate inaccuracy. The link layer is modeled as a discrete-time buffer queue with infinite buffer size. In nth time slot, we observe the arrival packet number A_n and departure packet number U_n. Based on the rate control policy $\mathcal{R}$, the departure number is

$$U_n = min(\mathcal{R}(g_n), X_{n-1} + A_n) \tag{5–33}$$

where X_{n-1} is the observed number of packets in the previous time slot. Then the following equation holds for the observed number of packets in nth time slot

$$X_n = max(X_{n-1} + A_n - U_n, 0) \tag{5–34}$$

If the time a packet stays in the buffer exceeds delay bound, it will be dropped. If a received packet cannot be correctly decoded at the receiver, it is considered lost. The

ratio of dropped packets plus the lost packets to the total number of packets is the average PEP. The simulations are carried out in three groups.

1.Given one video sequence, we keep the source coding distortion and rate (the number of packet) constant by keeping ME complexity parameters λ, θ and quantization parameter QP the same, while we change the encoding time $5 < T_e < 30$ by varying other parameters such as CPU power P in (5–12). We set end-to-end delay $T = 100ms$, and explore how encoding time T_e affect PEP. Figure 5-28 shows the packet drop probability due to delay bound violation increases with T_e in AWGN channel. In each time slot, we allow packet to transmit through the noisy channel at the rate of channel capacity proportion, with Gaussian noise 0 mean and $1/16$ variance. Intuitively, given the end-to-end delay, longer encoding time means shorter waiting time in encoder buffer, which leads to more packet drop. Figure 5-29 shows the PEP over AWGN channel, which has the same tendency as packet drop probability. Figure 5-30 and Figure 5-31 show the packet drop probability and PEP vs. T_e over wireless block fading channel. We set channel gain $\sqrt{g_n}$ of Rayleigh distribution with variance $\sigma^2 = 1$, white Gaussian noise the same as AWGN channel. Again, it shows packet drop probability and PEP increase with T_e given fixed T, if source distortion and rate are kept the same. It is observed for both channels, PEP is very close to packet drop probability in value throughout T_e, which indicates packet error mainly comes from packet drop rather than decoding error. This is because either the channel condition is quite good or the end-to-end delay is too tight in comparison to encoding time.

2. Given one video sequence, we change the encoding parameter T_e by varying ME complexity parameters, motion search range λ in this experiment, while keeping other parameters the same. It differs from simulation group 1 that source coding distortion and rate will change according to λ while T_e varies. Such a simulation is a better approximation to real video communication scenario where d-R-D control is achieved by tuning ME complexity parameter. We set search range $\lambda_0 = 4, 8, 16$,

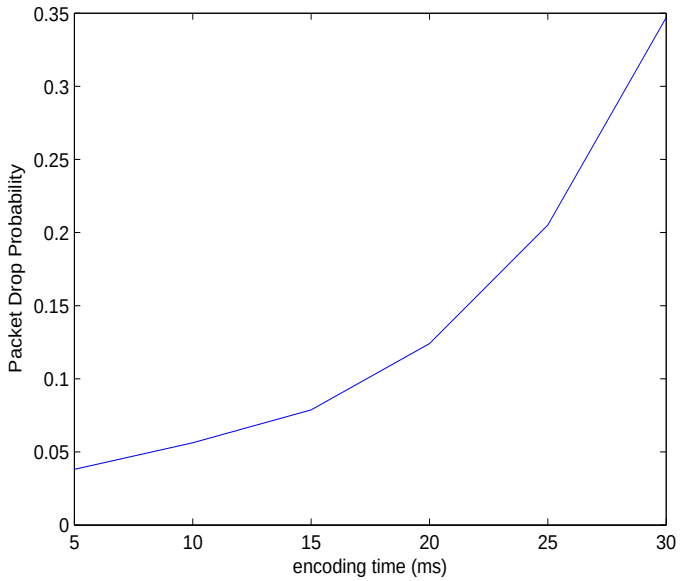


Figure 5-28. Packet drop probability vs. encoding time over AWGN channel given end-to-end delay for Foreman (qcif), keeping source distortion and rate the same.

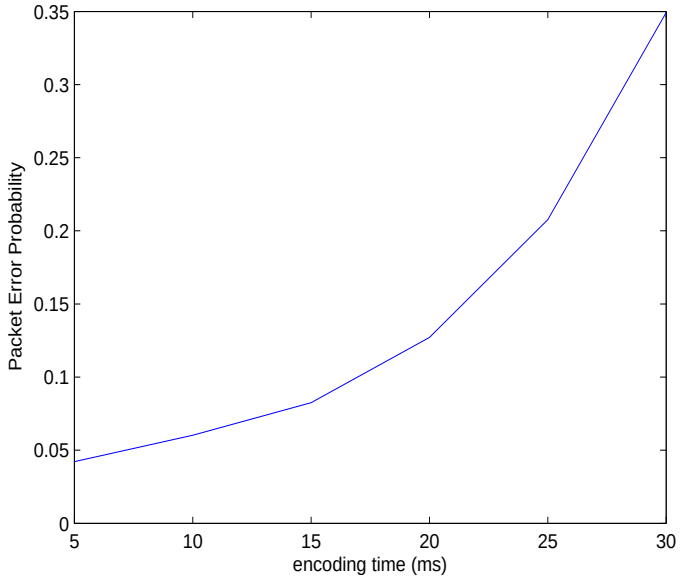


Figure 5-29. Packet error probability vs. encoding time over AWGN channel given end-to-end delay for Foreman (qcif), keeping source distortion and rate the same.

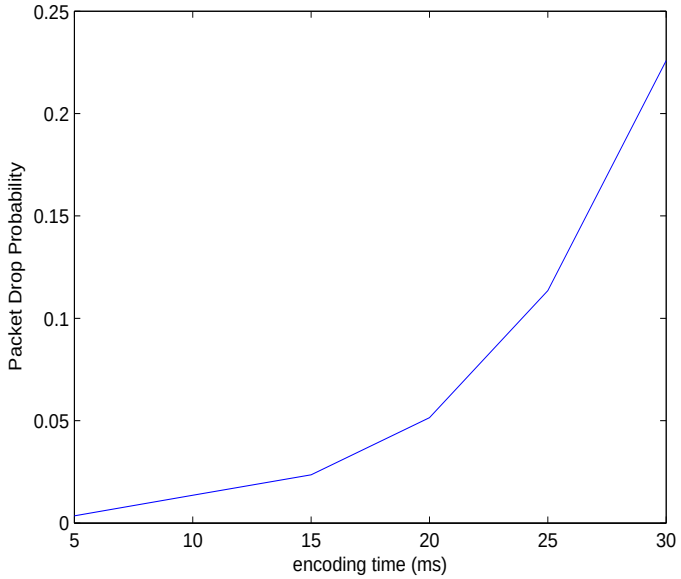


Figure 5-30. Packet drop probability vs. encoding time over block fading channel given end-to-end delay for Foreman (qcif), keeping source distortion and rate the same.

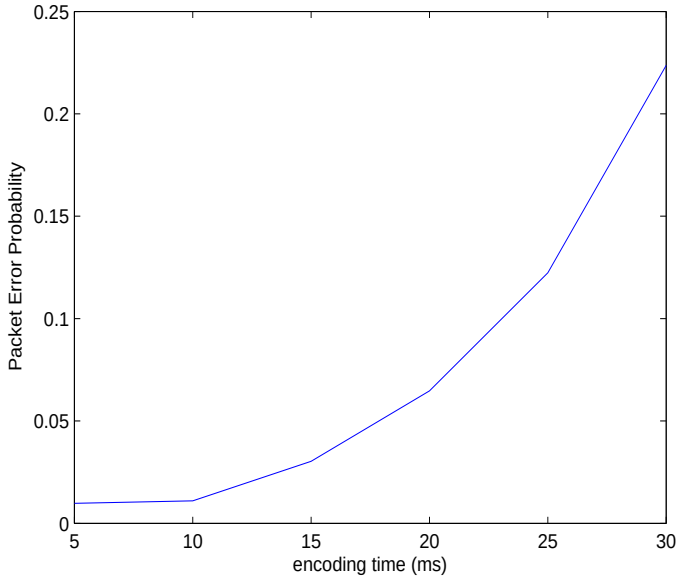


Figure 5-31. Packet error probability vs. encoding time over block fading channel given end-to-end delay for Foreman (qcif), keeping source distortion and rate the same.

Table 5-3. Packet Error with $T = 50ms$ over AWGN Channel of Different Encoding Time (ms) for Foreman

$T_e(\lambda_0)$	PEP	Average Dropped Packet	Average Erroneous Packet
2(4)	0.1236	1333	32.3
5(8)	0.0779	788	31.4
20(16)	0.1153	1160	38.5

Table 5-4. Packet Error with $T = 100ms$ over AWGN Channel of Different Encoding Time (ms) for Foreman

$T_e(\lambda_0)$	PEP	Average Dropped Packet	Average Erroneous Packet
2(4)	0.0564	582	40.5
5(8)	0.0030	0	31.4
20(16)	0.0038	0	39

and carefully select the other parameters in (5–12) to make $\lambda_0 = 4, 8, 16$ correspond to $T_e = 2, 5, 20ms$ respectively. Intuitively, small search range leads to more bitrate (number of packet) and inferior source quality, but it saves time to enable each packet to stay longer in encoding buffer, and vice versa. However, simulation results turn out to be more complicated than this simple intuition. Table 5-3 to 5-5 reveal PEP vs. T_e in 3 different end-to-end delay cases, $T = 50ms$, $T = 100ms$ and $T = 200ms$ over AWGN channel.

When end-to-end delay is tight ($100ms$) compared to encoding time, packet drop is inevitable, as shown in Table 5-3. Though the bitrate (number of packet) reduces due to elegant coding, $T_e = 20ms$ causes the available waiting time in encoding buffer too short, leading to large number of dropped packet (1160 out of 10393 packets). On the other hand, $T_e = 2ms$ makes up enough time in the buffer, but the rough ME causes

Table 5-5. Packet Error with $T = 200ms$ over AWGN Channel of Different Encoding Time (ms) for Foreman

$T_e(\lambda_0)$	PEP	Average Dropped Packet	Average Erroneous Packet
2(4)	0.0038	0	42.1
5(8)	0.0031	0	32.5
20(16)	0.0033	0	34.1

Table 5-6. Packet Error with $T = 50ms$ over Block Fading Channel of Different Encoding Time (ms) for Foreman

$T_e(\lambda_0)$	PEP	Average Dropped Packet	Average Erroneous Packet
2(4)	0.0076	36.3	48.2
5(8)	0.0041	41	49.3
20(16)	0.0260	234.5	35.6

Table 5-7. Packet Error with $T = 100ms$ over Block Fading Channel of Different Encoding Time (ms) for Foreman

$T_e(\lambda_0)$	PEP	Average Dropped Packet	Average Erroneous Packet
2(4)	0.0039	0	43.3
5(8)	0.0041	0	43.5
20(16)	0.0037	0	38.2

more bitrate and hence more packets to process, which also creates a large number of dropped packet (1333 out of 11046 packets). In Table 5-4, the end-to-end delay is more relaxed compared with encoding time, and $T_e = 5ms$ or $T_e = 20ms$ does not have any difference in packet drop rate (all reduce to 0). Therefore, the number of packets to process plays an important role in the overall PEP. As rough coding of $T_e = 2ms$ results in more bitrate than $T_e = 5ms$ and $T_e = 20ms$, it has the most significant packet drop among the 3 cases. If end-to-end delay continues to increase to $200ms$, the importance of encoding time becomes negligible in the overall PEP. As demonstrated in Table 5-5, PEP only comes from decoding error for all 3 cases, with no packet drop at all! Under such circumstance, the one with the least source distortion ($T_e = 20ms$) wins over others, given the same error concealment strategy at the decoder. Table 5-6to 5-8 present PEP vs. T_e in 3 different end-to-end delay cases, $T = 50ms$, $T = 100ms$ and $T = 200ms$ over block fading channel. It is not surprising that all packet error statistics in block fading channel have the same characteristics as AWGN channel.

Summarize the results in all tables, we see that encoding time $T_e = 5ms$ achieves the best trade-off between source distortion (measured in T_e) and transmission distortion (measured in PEP) among all 3 end-to-end delay cases. We draw a

Table 5-8. Packet Error with $T = 200ms$ over Block Fading Channel of Different Encoding Time (ms) for Foreman

$T_e(\lambda_0)$	PEP	Average Dropped Packet	Average Erroneous Packet
2(4)	0.0032	0	35.7
5(8)	0.0044	0	46.8
20(16)	0.0035	0	36.5

conclusion that medium encoding time compared with end-to-end delay should be preferred. In real time video applications like video conference, the tolerable end-to-end delay is often around $250ms$, while the maximum encoding time for each frame is only $33ms$, hence $T_e \ll T$. Consequently, it is acceptable to neglect the encoding time as is assumed in most works [78][79][82]. However, if the encoding time for each frame increases, for example high frame rate, or the ratio of T_e to T is tight, it is not reasonable to ignore the encoding time in end-to-end delay analysis.

3.Given one video sequence, we change the encoding parameter T_e by varying ME complexity parameters, number of reference frame θ in this experiment, while keeping other parameters the same. We set $\theta = 1, 2, 4, 6$, and carefully select the other parameters in (5–12) to make $\theta = 1, 2, 4, 6$ correspond to $T_e = 5, 10, 20, 30ms$ respectively.The simulation results are illustrated in Table 5-9 to 5-11. When θ increases, encoding time T_e also increases, and the available encoding buffer time T_{eb} decreases given end-to-end delay. Because increasing θ does not reduce much of bitrate and thus the number of packet, PEP is largely determined by packet drop probability, where small θ has smaller packet drop probability. This is true for both short and long end-to-end delays. Note $T = 200ms$ is an exception here, since packet drop probability reduces to 0 for all θ, and they should have the same PEP if over the same AWGN channel, which has been proved in Table 5-11.

Although the simulations in this section demonstrate the encoding time influence on the overall end-to-end video transmission system performance, a analytical model is necessary for developing the end-to-end rate-distortion optimization framework in

Table 5-9. Packet Error with $T = 50ms$ over AWGN Channel of Different Encoding Time (ms) (Reference Frame Number) for Foreman

$T_e(\theta)$	PEP	Average Dropped Packet	Average Erroneous Packet
5(1)	0.0108	80	50.5
10(2)	0.0123	106	32.9
20(4)	0.0499	489	35.8
30(6)	0.1739	1750	27.6

Table 5-10. Packet Error with $T = 100ms$ over AWGN Channel of Different Encoding Time (ms) (Reference Frame Number) for Foreman

$T_e(\theta)$	PEP	Average Dropped Packet	Average Erroneous Packet
5(1)	0.0029	0	35.4
10(2)	0.0040	0	35.4
20(4)	0.0039	0	40.7
30(6)	0.0040	0	41.3

order to choose the optimal encoding time, either on frame level or sequence level. A few points should be taken into account when developing such a model. Assume propagation delay and decoding time constant, end-to-end delay is composed of encoding time and encoding buffer delay. The input of the encoding output buffer is the coded bitstream packets with arrival rate not necessarily being constant. The output of the buffer enters channel, and the departure rate depends on current channel condition. The channel is time-variant, and a proper channel model could be block fading channel as in the simulation. Channel coding is employed to protect the video packets[114][115]. Transmission distortion is a result of packet drop and erroneous transmission, wherein packet drop includes both delay bound violation drop and

Table 5-11. Packet Error $T = 200ms$ over AWGN Channel of Different Encoding Time (ms) (Reference Frame Number) for Foreman

$T_e(\theta)$	PEP	Average Dropped Packet	Average Erroneous Packet
5(1)	0.0041	0	50.1
10(2)	0.0028	0	31.8
20(4)	0.0036	0	38.1
30(6)	0.0040	0	40.5

packet overflow drop [84][90][112]. The source coding distortion and transmission distortion are additive to derive the end-to-end distortion. In summary, we have to develop a delay-rate-distortion model for video transmission, where ”delay“ indicates encoding buffer delay, ”rate“ indicates channel bit-rate, and ’distortion“ is transmission distortion. None of the related work analyze all three dimensions at the same time [116][117][118][119][120][121]. It is our future work to develop a d-R-D model for wireless channel transmission.

5.2 Model of Source Video Coding in Hierarchial-B Coding

5.2.1 Hierarchical B Pictures Coding

In section 5.1.2, we developed d-R-D model for IPPPP coding mode, and the encoding time or d in d-R-D model is a frame based measurement, i.e. encoding time for a single P frame. This model is only appropriate for delay sensitive applications such as video conference and real time video streaming, where the tunable delay range is very limited. For example, if the frame rate is 30 fps, then the maximum encoding time for each frame is the frame duration of $33ms$. However, for delay insensitive video communication, the delay constraint is more flexible to use coding mode other than IPPPP. Even the live television broadcast is able to have a delay of around one second, not to mention the none-live TV broadcast. These delay insensitive applications usually employ B frames. By using bidirectional prediction from both past and future frames, compression performance improves because the temporal correlation among several neighboring frames is exploited, but additional delay is incurred [100]. Recently, a hierarchical bi-predictive structures called hierarchical B pictures is adopted in H.264/AVC, which is originally proposed to scalable video coding in motion compensated temporal filtering (MCTF) extension. Analysis has shown that the major part of the reported gain comes from the hierarchical coding structure, not from the additional motion compensated update steps in the MCTF extension [122].

A typical hierarchical B picture coding structure with 4 dyadic temporal levels is illustrated in Figure 5-32. More often than not, only the first frame is intra-coded as IDR (instant direct refresh) picture, The key picture is coded at a regular interval, 8 in 4 dyadic temporal levels here, and can be either inter-coded (P frame) using previous key pictures or intra-coded (I frame) to allow random access. A key picture and the all pictures temporally located between the key frame and the previous key frame creates a group of picture (GOP). Any frame within a GOP are hierarchically coded as B frames. For example in Figure 5-32, frame 1 to 8 builds one GOP, and frame 8 is the key picture, inter-coded as P frame by predicting from the only IDR picture frame 0, denoted as level 0. Frame 4 is bi-predicted from both frame 0 and 8 and denoted as level 1; frame 2 is then bi-predicted from frame 0 and 4, and frame 6 bi-predicted from frame 4 and 8, and these two frames are denoted as level 2; finally, frame 1, 3, 5, 7 are bi-predicted from each of their adjacent frame in both directions, and denoted as level 3. This builds a 4 level hierarchical structure, and the coding order follows the level order, from the lowest to the highest. It is obvious that such a hierarchical coding structure introduces large delay, since the video input order differs from the coding order. It is impossible to start to code the key picture before the inputs of the entire GOP finishes. Consequently, the measurement of encoding time in hierarchical B pictures should be in the size of a GOP N_{gop}, not an individual frame duration as in IPPPP mode. Specifically, if a GOP is of size n, and the input frame interval is $T_f = 1/f$ (f is frame rate), the encoding time is

$$T_e = n \times T_f \tag{5–35}$$

which is way more larger than encoding time of a single frame based on ME complexity as in IPPPP mode in section 5.1.2. In the next section, we will develop delay-rate-distortion source coding model for hierarchical B pictures, and GOP size N_{gop} is defined to be the complexity control parameter, i.e. delay is a function of N_{gop}, $d(N_{gop})$. Recall that in d-R-D model for IPPPP mode, we exploit the relationship between source

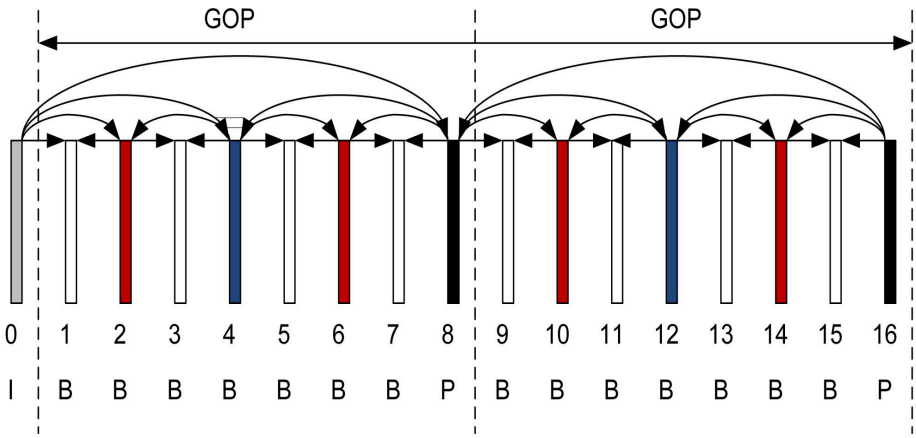


Figure 5-32. Dyadic hierarchical coding structure with 4 temporal levels and a GOP size of 8. Arrow indicates prediction direction.

coding distortion, rate and ME complexity parameters λ, θ. We will adopt the similar strategy in hierarchical B pictures case, trying to develop D and R as functions of N_{gop}.

5.2.2 Model of Source Coding in Hierarchical-B Mode

Recall that in d-R-D model for IPPPP coding mode, we investigate the relationship between residual variance and ME complexity parameters through experiments so as to relate D and R with d, as in (5–11) and (5–12). Similarly, we borrow the function form of D in [89] and R in [90], and explore the functional relation between residue variance and GOP size N_{gop}, i.e.

$$\begin{aligned} D(Q,\sigma) &= D(Q,\sigma(N_{gop})) \\ R(Q,\sigma) &= R(Q,\sigma(N_{gop})) \\ & \quad d(N_{gop}) \end{aligned} \tag{5–36}$$

In hierarchical B pictures encoding, a GOP is a composed of a hierarchy of several layers. For a better explanation, Figure 5-33 depicts a dyadic hierarchical B structure of 3 levels with GOP size 4. Level 0 is inter-coded key picture directly predicted from the first IDR picture. Level 1 is bi-predicted B frame from the neighboring two key frames, and this B frame is used as reference (RefB) for the next level. Level 2 includes two

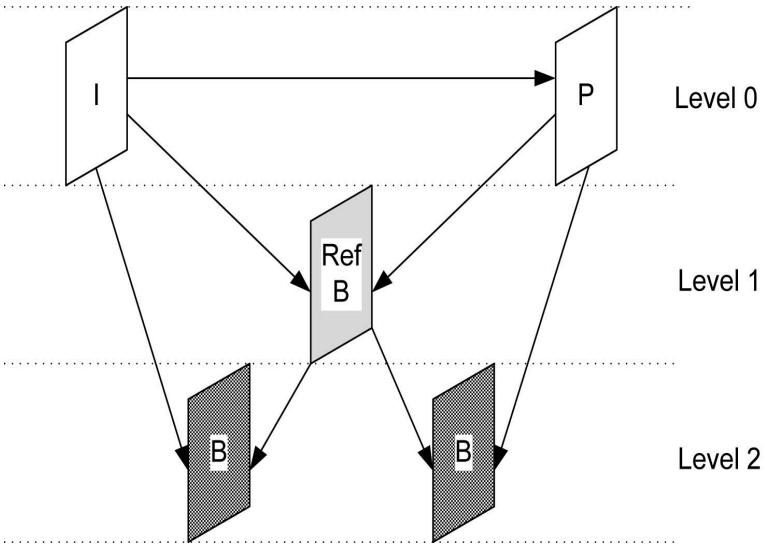


Figure 5-33. Dyadic hierarchical coding structure with 3 temporal levels and a GOP size of 4.

none reference B frames, bi-predicted from one reference in Level 0 and the RefB in level 1. To relate predictive residual variance with GOP size, we have to consider two aspects in this hierarchical B structure. 1)The key picture and the remaining pictures of the GOP should be considered separately. The key picture is an inter-coded P frame, with one directional prediction from the previous key frame (IDR frame if the first GOP), while the remaining pictures are bi-predictive B frames. It has been analyzed in [100] and [123] that two hypotheses are always better than a single hypotheses in sense of predictive distortion. We can infer that the functional form of σ and N_{gop} may differ in P and B frame. Even if they are of the same functional form, it is very likely the function parameters for P and B frame are different. 2)Apart from ME complexity parameters such as λ and θ, motion compensation accuracy may depend on the temporal distance between the reference and the predicted frame. This is particularly true in hierarchical B structure. Taken dyadic hierarchical B structure with 3 and 4 temporal levels as an example, in Figure 5-32, the temporal distance between key picture is 8, while in Figure 5-33, the temporal distance reduces to 4. We will take a closer look at how this distance will affect motion compensation accuracy and thus residual variance.

In previous work [100], it proposed the standard deviation of the motion compensation displacement error σ_Δ varies approximately linearly with the temporal prediction distance Δ_t [100]. However, this proposal has only been verified for single hypotheses prediction, i.e. P frame. Furthermore, this conclusion is drawn from the logarithmically spaced rate-distortion function given different temporal distance, rather than a explicitly linear function proof between σ_Δ and Δ_t. Finally, it fails to estimate the linear model parameters adaptively for different video content, but rather keep them constant for all sequences as in [124]. We will solve these problems by establishing the linear functional relationship for P frame, and investigate this function for B frame as well.

5.2.2.1 Variance vs. temporal predictive distance for P frame

We encoded several video sequences for varying temporal prediction distance. The different temporal prediction distances are obtained by varying the temporal sub-sampling ratio [100]. The results of two sequences Foreman (qcif) and RaceHorses (WVGA) are shown in Figure 5-34. We use IPPPP coding mode, and original frame rate is 30 fps. Hence, 30 fps corresponds to temporal distance $\Delta_t = 1$, sub-sampling ratio of 2 leads to 15 fps , corresponding to $\Delta_t = 2$, a sub-sampling ratio of 3 gets 10 fps, corresponding to $\Delta_t = 3$, so and and forth. The derived average residual variance with varying Δ_t does resemble linear function, and we do linear fitting based on the following equation which has been proposed in [100]

$$\sigma(\Delta_t) = \alpha + \beta(\Delta_t - 1) \tag{5–37}$$

It is seen from Figure 5-34 that the linear fitting is quite accurate to the true residual variance.

5.2.2.2 Variance vs. temporal prediction distance for B frame

We know that in hierarchical B pictures coding, B frame is predicted from two hypotheses, one from past and one from future, and the two predictions are weighted

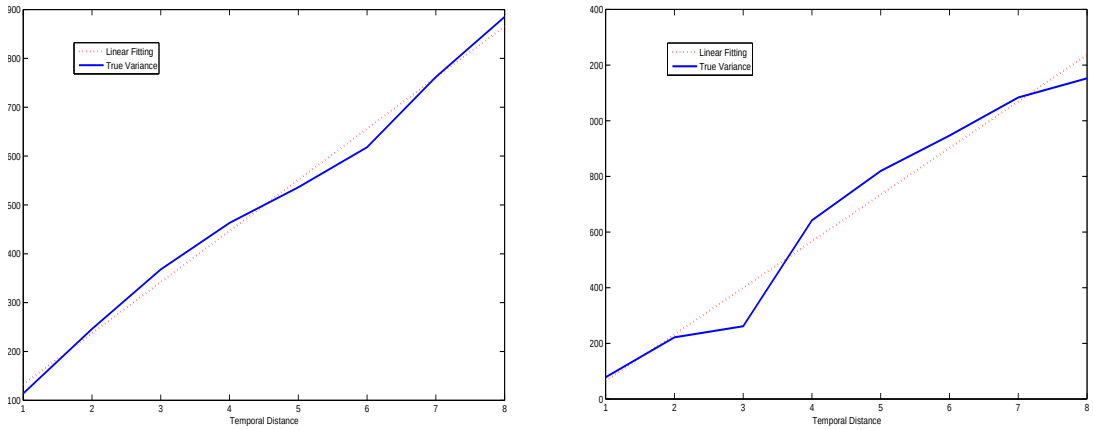
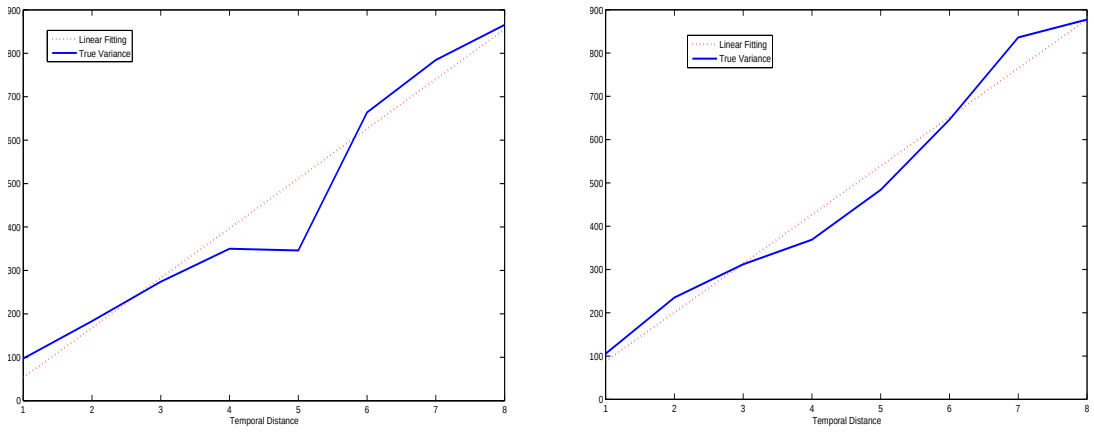


Figure 5-34. Linear fitting of true residue variance vs. temporal distance for single hypothesis prediction (P frame). (a)Foreman (qcif). (b)RaceHorses (WVGA).

to form the prediction for this B frame. Because of the hypotheses number difference, it is likely the relationship between motion compensation accuracy and temporal distance for B frame differs from that for P frame, but no results haven been reported concerning this to the best of our knowledge. Therefore, we will establish the relationship between σ and Δ_t in this section. Similar to the strategy we use in section 5.2.2.1, we vary the frame rate of the video sequence by varying the sub-sampling ratio. And we apply IBPBP coding mode in H.264/AVC, so that it includes the B frame we want to study, while the hypotheses only come from intra or inter-coded P frame without any RefB. Also, we set QP for P 28, and QP for B 30 in all temporal distances to mitigate the effect of quantization distortion on the obtained results. The data of average residual variance under varying temporal distance is given in Table 5-12, which is also close to linear shape. To unify $\sigma(\Delta_t)$ of P and B frame, we adopt the linear model in (5–37), and fitting results for the same two sequences are shown in Figure 5-35. It is observed that the residual variance varying temporal distance for B frame generally follows a linear shape, but the fitting parameters α and β are no way close to parameters for P frame. Hence,

Table 5-12. Residual Variance σ vs. Temporal Distance Δ_t of B frame in Foreman (qcif)

Δ_t	1	2	3	4	5	6	7	8
σ	97.17	183.28	274.03	350.04	346.04	664.03	784.70	465.47



A B

Figure 5-35. Linear fitting of true residue variance vs. temporal predictive distance for two hypotheses prediction (B frame). (a)Foreman (qcif). (b)RaceHorses (WVGA).

to compromise the parameters for both P and B frames and all sequences as in [100] is impossible. We will estimate the fitting parameters for P and B separately in this work.

5.2.2.3 d-R-D model for GOP in hierarchical-B pictures

Let l denote the temporal decomposition level, the temporal distance with respect to the level and the GOP size N_{gop} can be written as [100]

$$\Delta_t(l) = 2^{-l} N_{gop}, l \in [0, L] \tag{5–38}$$

We assume dyadic hierarchical B structure, and GOP size is always $N_{gop} = 2^L$. For example, in a dyadic hierarchical structure with 2 temporal levels in Figure 5-33, $N_{gop} = 2^2 = 4$, when $l = 0$, P frame has $\Delta_t = N_{gop} = 4$, when $l = 1$, RefB has $\Delta_t = 2$, and when $l = 2$, the remaining B frames have $\Delta_t = 1$. Because the mapping of N_{gop} and temporal distance is based on level, every frame in a GOP has individual σ that can be plugged into D, R functions in (5–36). To derive a uniform D and R as functions of N_{gop}, we have to define D and R as average coding distortion and rate for an entire GOP. This

can be easily achieved by calculating distortion and rate for each frame in the GOP, with $\sigma(\Delta_t(l))$ for that particular frame (either P or B frame), and averaging over the GOP size. Specifically,

$$D(QP, N_{gop}) = \frac{1}{N_{gop}} \sum_{i=1}^{N_{gop}} D(QP(i), \sigma(\Delta_t(l(i)))) \tag{5–39}$$
$$R(QP, N_{gop}) = \frac{1}{N_{gop}} \sum_{i=1}^{N_{gop}} R(QP(i), \sigma(\Delta_t(l(i))))$$

where both QP and σ are functions of an individual frame in the GOP. It has been explained above σ is determined by the temporal distance of every frame, which depends on the hierarchical level that frame is located. On the other hand, QP can differ for individual frame in two ways: 1)different frame type is often assigned different QP, such as $QP(B) = QP(P) + 2$ in H.264/AVC configure. 2)different hierarchical levels can be assigned different QP. Many simulations have reported that the modification of QP for B frame in different decomposition levels can improve the coding efficiency [122][125]. For example, it shows $QP(B_n) = QP(B_{n-1}) + 1$ improves the coding performance than $QP(B) = QP(P) + 2$. Recall that we define the encoding time in hierarchical B pictures a function of GOP size

$$d(G_{gop}) = N_{gop} \times T_f \tag{5–40}$$

Combining (5–39) and (5–40), we have established the d-R-D model of hierarchical B pictures, with tuning parameter N_{gop}. Like in IPPPP coding, we still need to verify the model accuracy by comparing the model estimate with the true R-D.

We set the coding configurations in H.264/AVC software JM16.0 to enable hierarchical B pictures, and vary the GOP size from 2 to 16 Note currently, the maximum GOP size of hierarchical B pictures supported in H.264/AVC is 32. In sequence Foreman (qcif), QP is used the traditional setting,$QP(B) = QP(P) + 2$; for RaceHorses (WVGA) we modified the QP settings to $QP(B_n) = QP(B_{n-1}) + 1$, with

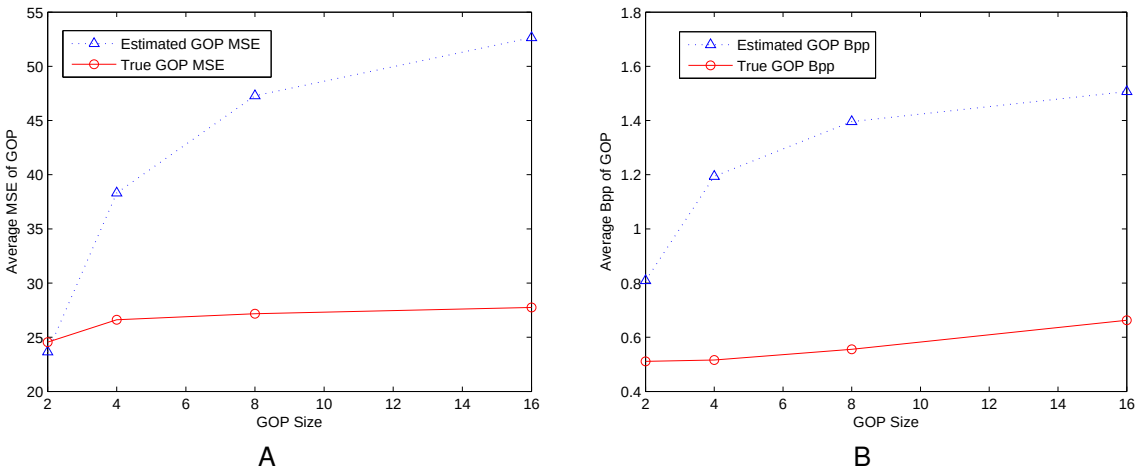


Figure 5-36. Model estimate of D and R vs. GOP size in Foreman (qcif) in hierarchical B pictures, in comparison to true coding value. (a)GOP distortion (MSE). (b)GOP bitrate (Bpp).

$QP(B_1) = QP(P) + 2$, in order to test if the model can follow the QP change accordingly. We keep the true coding distortion in MSE and coding rate in Bpp of individual frame, and average over the GOP size, compare the average distortion and rate with the model estimate. The results gained are presented in Figure 5-36 and Figure 5-37 for the two sequences. Obviously, the model estimate is not very accurate, due to the following reasons. 1) D and R as functions of residual variance σ and quantization parameter QP have mismatch themselves. This has been confirmed in section 5.1.3 through extensive experiments. 2) The motion compensation accuracy of level n is dependent on distortion of level $n-1$ in hierarchical B structure. However, D and R model of GOP (5–39) only take individual frame D and R and average over the GOP size, without considering the level dependency.

Being aware that the problem comes from the deficiency of the model itself, we will try to improve the model accuracy by supportive method. We choose to adopt compensation scheme as in (5–14) and (5–15), just revise the compensate unit from frame to GOP. Figure 5-38 and Figure 5-39 show the compensated D and R model estimate of the same two sequences, comparing with the true coding value as long as

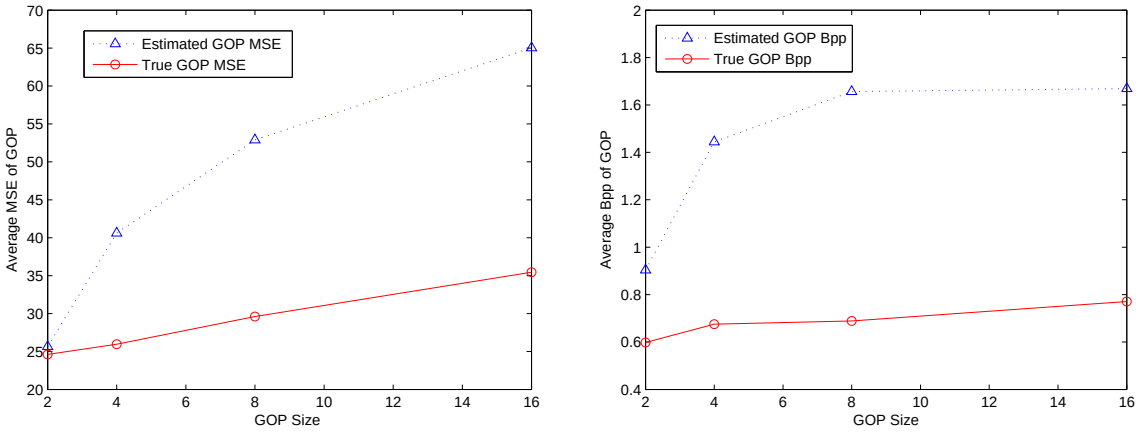


Figure 5-37. Model estimate of D and R vs. GOP size in RaceHorses (WVGA) in hierarchical B pictures, in comparison to true coding value. (a)GOP distortion (MSE). (b)GOP bitrate (Bpp).

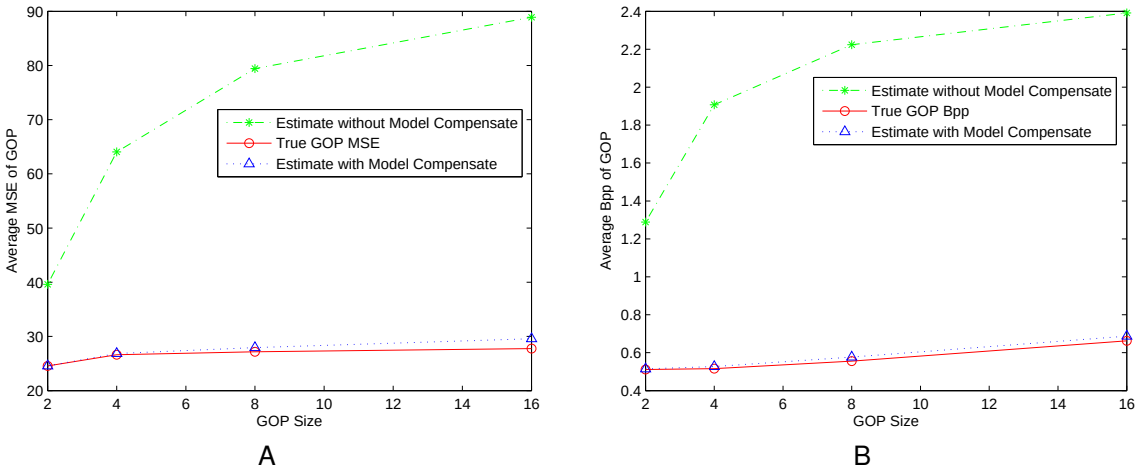


Figure 5-38. Compensated model estimate of D and R vs. GOP size in Foreman (qcif) in hierarchical B pictures, in comparison to true coding value and uncompensated model estimate (a)GOP distortion (MSE). (b)GOP bitrate (Bpp).

the uncompensated estimate. From the figure, we see that the compensated model has greatly improved the estimate accuracy than the uncompensated model.

5.2.3 Rate Distortion Optimized Source Rate Control in Hierarchical-B Structure

In section 5.2.2.3, we noticed that in experiments, when GOP size of hierarchical B structure increases, the distortion and bit-rate also increases, indicating larger GOP

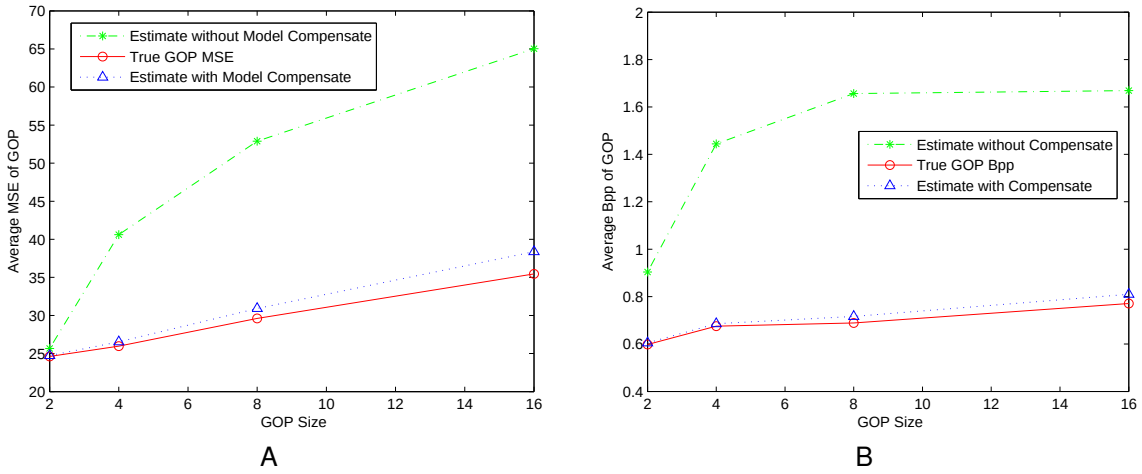


Figure 5-39. Compensated model estimate of D and R vs. GOP size in RaceHorses (WVGA) in hierarchical B pictures, in comparison to true coding value and uncompensated model estimate (a)GOP distortion (MSE). (b)GOP bitrate (Bpp).

size has worse coding efficiency. In fact, this is not always the case. [122] and [100] all show results that support the opposite situation, i.e. larger GOP size increases the coding efficiency. This is because we manually set constant quantization parameters QP for P and B frames in our simulations, while [122][100] employ rate control that enables QP changes for individual P and B frames in all GOP size. Once we enable rate control for hierarchical B structure in JM16.0, we also see improved coding efficiency when GOP size increases. Such a result is not difficult to understand. Since some B frames are not used for reference to predict other frame, the quantization parameter can be greater than its adjacent P or I frames such that the bits could be saved for I and P frames [126]. On the other hand, it is proved in [123] that estimate from two references (bi-directional prediction as B frame) is always superior to from one reference (forward prediction as P frame), so B frame will consume less bits than P frame given the same QP. This is particular advantageous in case of global motion and camera pan [127]. However, to keep increasing GOP size does not show monotonous increase of coding efficiency. This is especially true for fast moving sequences, in that large GOP size also increases the prediction distance which leads to poorer prediction and needs more bits.

Therefore, it is a tradeoff between saving more bits with more number of B frames and consuming more bits with longer predictive distance when GOP size increases. And how to adaptively determine GOP size so as to optimize coding efficiency becomes a challenging problem in hierarchical B coding.

The GOP size N_{gop} is the parameter of structure delay, which is non-trivial in hierarchical prediction system. Some previous works have studied the effect of structure delay interrelated into hierarchical prediction structure. A study on reduced structural delay appeared for MCTF systems appears in [128]. Tradeoffs of delay and compression in MCTF video codecs were investigated in [129]. Different from hierarchical B coding, MCTF is a open loop approach, in which B-coded pictures are predicted from original reference rather than reconstructed ones. Recently, the update step was removed from the working draft of the Scalable Video Coding extension to H.264/AVC [130][131][132]. For details of MCTF, readers can refer to [133][134][135][136]. [137] and [100] touches the effect of branch removal in order to change the hierarchical B picture coders. Meanwhile, [100] studies the tradeoff of compression efficiency for delay in hierarchical B structure. However, the delay considered is mainly due to buffering at the encoder output instead of structure delay, and it translates the end-to-end delay constraint into rate allocation constraint. To the best of our knowledge, no existing literature ever mathematically analyzes the relationship between compression efficiency and structure delay in hierarchical-B coding. In this section, we apply the proposed d-R-D model for GOP to determine structure delay parameter N_{gop}, so that rate distortion of the average GOP in hierarchical B coding can me optimized.

Note here we address the criterion of ”average“ distortion and bit-rate of a GOP because the number of picture changes with GOP size, and the coding order of a single frame also changes in different GOP size, so it does not make sense to compare the picture quality of each single frame. On the other hand, GOP is the basic coding unit in the rate distortion optimization, and the target is to minimize the average GOP distortion

within bit-rate constraint. Accordingly, the rate distortion optimization over average GOP problem can be formulated below:

$$\begin{aligned} min \quad & D_{gop}(QP, \sigma(N_{gop})) \\ s.t \quad & R_{gop}(QP, \sigma(N_{gop})) \leq R_0 \\ & N_{gop} \leq N \end{aligned} \tag{5–41}$$

where D_{gop} is the average distortion of a GOP of size N_{gop} and can be rewritten as

$$D_{gop}(QP, \sigma(N_{gop})) = \frac{1}{N_{gop}} \sum_{i=1}^{N_{gop}} D_i(QP_i, \sigma(\Delta_t(I(i)))) \tag{5–42}$$

while R^{gop} is the average bit-rate of a GOP of size N_{gop} and can be rewritten as

$$R_{gop}(QP, \sigma(N_{gop})) = \frac{1}{N_{gop}} \sum_{i=1}^{N_{gop}} R_i(QP_i, \sigma(\Delta_t(I(i)))) \tag{5–43}$$

σ is determined for every single frame in a GOP depending on the layer location of that frame. N is the maximum number of frames in a GOP to be within the structure delay as in (5–40), or to meet the GOP size limit of a particular encoder. For example, H.264/AVC codec JM16.0 supports maximum GOP size of 32, i.e. $N = 32$. We may use Lagrangian approach to solve (5–41) as

$$\begin{aligned} min \quad & \frac{1}{N_{gop}} \sum_{i=1}^{N_{gop}} D_i(QP_i, \sigma(\Delta_t(I(i)))) + \Lambda \frac{1}{N_{gop}} \sum_{i=1}^{N_{gop}} R_i(QP_i, \sigma(\Delta_t(I(i)))) \\ s.t \quad & N_{gop} \leq N \end{aligned} \tag{5–44}$$

A practical algorithm of RDO to find the optimal GOP size in hierarchical B and quantization parameter of each frame in the GOP is described in detail below.

Algorithm 4: GOP level rate-distortion optimized determination of GOP size N_{gop} and quantization parameter QP.

set $J_{min} = inf$

for $\Lambda_j = \Lambda_0, \Lambda_1, \cdots, \Lambda_N$ **do**

for $N_{gop} \leq N$ **do**

for $i \leq N_{gop}$ **do**

for $QP_{min} \leq QP_i \leq QP_{max}$ **do**

estimate D_i by (5–38), (5–37), (5–7) and (5–6),

estimate R_i by (5–38), (5–37), (5–7) and (5–10),

end for

calculate $J = \frac{1}{N_{gop}} \sum_{i=1}^{N_{gop}} D_i + \Lambda_j \frac{1}{N_{gop}} \sum_{i=1}^{N_{gop}} R_i$,

if $J < J_{min}$ **then**

$J_{min} = J$,

$N_{gop}(\Lambda*) = N_{gop}$,

$QP_i(\Lambda*) = QP_i$,

$\lambda* = \lambda_j$,

end if

end for

end for

end for

The loop to find the optimal QP for each frame is extremely time consuming when $N_{gop} > 8$. To speed up the computation, we modify Algorithm 4 by introducing early termination in loop when $N_{gop} > 8$ as in Algorithm 5.

Algorithm 5: GOP level rate-distortion optimized determination of GOP size N_{gop} and quantization parameter QP with early termination.

set $J_{min} = inf$

for $\Lambda_j = \Lambda_0, \Lambda_1, \cdots, \Lambda_N$ **do**

for $N_{gop} \leq N$ **do**

for $i \leq N_{gop}$ **do**

if $N_{gop} \leq 8$ **then**

$QP_L = QP_{min}$

else

$QP_L = QP_{l-1}$

end if

for $QP_L \leq QP_i \leq QP_{max}$ **do**

estimate D_i by (5–38), (5–37), (5–7) and (5–6),

estimate R_i by (5–38), (5–37), (5–7) and (5–10)

if $N_{gop} > 8$ **then**

if $\sum_{i=1}^{i} R_i > R_0^i$ **then**

terminate the loop of QP_i,

end if

end if

end for

calculate $J = \frac{1}{N_{gop}} \sum_{i=1}^{N_{gop}} D_i + \Lambda_j \frac{1}{N_{gop}} \sum_{i=1}^{N_{gop}} R_i$,

if $J < J_{min}$ **then**

$J_{min} = J$,

$N_{gop}(\Lambda*) = N_{gop}$,

$QP_i(\Lambda*) = QP_i$,

$\lambda* = \lambda_j$,

end if

end for

end for

end for

In Algorithm 5, we restrict the lower bound of QP for current frame located at layer l to be no smaller than the QP of frame from previous layer $l-1$. For one thing, this is based on consideration that frame in layer l has shorter predictive distance than layer l, hence residue signal is generally smaller for shorter distance. For the other, frames

from upper layer are more likely to be used for references, such as P frame in layer 0, and the first B frame in layer 1 when $N_{gop} = 4$, so on and so force. The cascading of quantization parameter in hierarchical B structure has been presented in [122], which shows improved performance than uniform QP for all layers. Besides, the similar idea to use larger QP for B frame than P frame has been specified in H.264/AVC rate control, where quantization for B pictures is selected to be equal to quantization parameter of I/P pictures plus 2 [138][126][139]. Moreover, for early termination, we constantly check the sum of bit consumption of all currently looped frames in the GOP. If the bit usage exceeds the total available bits for the entire GOP, the selected QP combination for currently looped frames is disqualified and hence we terminate the loop.

We apply Algorithm 5 with early termination to two test sequences Foreman (QCIF) and RaceHorses (WVGA), to find the best GOP size and quantization parameter combination in hierarchical B structure coding. Assume the maximum GOP size $N = 16$, and input the fitted linear model (variance vs. predictive distance) parameters for P and B frame from section 5.2.2.1 and 5.2.2.2 respectively for each sequence. For Foreman, RDO selects $N_{gop} = 16$ as the optimal GOP size, with QP combination of $\{QP_0 = 30, QP_1 = 30, QP_2 = 33, QP_3 = 32, QP_4 = 32, QP_5 = 33, QP_6 = 32, QP_7 = 32, QP_8 = 32, QP_9 = 32, QP_{10} = 32, QP_{11} = 32, QP_{12} = 32, QP_{13} = 32, QP_{14} = 32, QP_{15} = 32\}$, where QP_0 is the quantization parameter for the only P picture in a GOP, and the rest are quantization parameters for B pictures in display order. We compare the RDO selected QP combination with other two schemes at $N_{gop} = 16$ in hierarchical B coding within the rate constraint $200Kbps$: 1)uniform QP for all frames in GOP as $QP = 33$; 2)cascading QP for B picture with increment of 1 in adjacent layers, $QP(k) = QP(k) + 1$. A single intra picture is used at the beginning of each sequence, and we set quantization parameter for intra picture the same for all three schemes $QP = 27$ to guarantee fair comparison. The result is illustrated in Figure 5-40. The RDO selected QP achieves the best overall performance at $PSNR = 33.330dB$, the QP cascading scores the second

Table 5-13. Average Y-PSNR(dB) of GOP in Foreman (QCIF) at 200Kbps under different GOP size

N_{gop}	16	2	4	8
$PSNR$	33.0106	32.9901	32.5695	32.3722

with $PSNR = 33.322dB$, and the uniform QP is the last with $PSNR = 33.246dB$. Notice that QP cascading scheme obtains better quality in some frames than RDO. However, due to the large quality fluctuation, its overall performance is inferior to our proposed RDO. Another interesting comparison could be with a rate distortion analysis only on $N_{gop} = 16$ similar to the strategy in [140], which will be our future work.

Intuitively, larger GOP size grants more freedom of bit allocation to each frame within this GOP. A good rate control approach should assign more bits to frames used for future references, or frames with large predictive residue. This is particularly true for sequence with slow motion content, in which the predictive distance will not lead to big residue signal due to motion or scene change when GOP size increases. On the other hand, picture of small resolution is more tempted to choose large GOP size as reference degradation due to increasing predictive distance is not that obvious in low bitrate application. Therefore, rate distortion optimization choose $N_{gop} = 16$ for Foreman (QCIF) as the optimal GOP length. The other selections that follow are listed in Table 5-13. On the contrary, RaceHorses (WVGA) chooses the optimal GOP size $N_{gop} = 2$ in RDO, because RaceHorses content has more fast motion, and the resolution ($412x240$) is much bigger than Foreman as well. The optimal quantization combination for $N_{gop} = 2$ is $\{QP_0 = 33, QP_1 = 34\}$. Here we do not compare the proposed RDO with other two QP combination because 1)the optimal QP combination is a cascading scheme itself; 2)for $N_{gop} = 2$, uniform QP for P/B pictures has only two available choice, that will lead to either better quality ($QP = 33$) or worse quality ($QP = 34$), plus $QP = 33$ for P/B pictures will break bit-rate constraint. Table 5-14 lists all the selections, with bitrate constraint $800Kbps$.

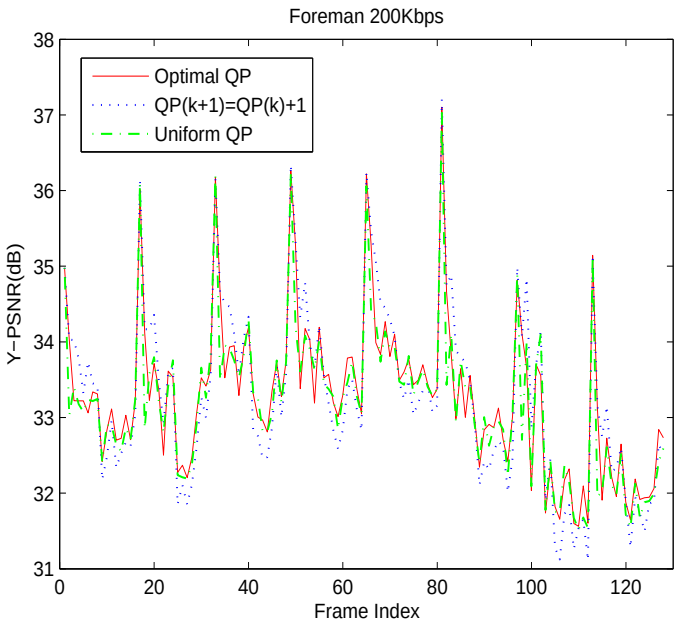


Figure 5-40. Picture quality of Foreman with rate control at 200Kbps, hierarchical B structure size 16 in H.264/AVC over different QP settings. Optimal, $PSNR = 33.330dB$, incremental $PSNR = 33.322dB$, uniform $PSNR = 33.246dB$.

Table 5-14. Average Y-PSNR(dB) of GOP in RaceHorses (WVGA) at 800Kbps under different GOP size

N_{gop}	2	16	4	8
$PSNR$	31.8388	30.7210	30.7857	30.2325

5.3 Summary and Future Work

In this chapter, we propose a source coding delay-Rate-Distortion model for the first time to study the relationship among the three coding variants. Due to the different delay constraint in real-time and none real-time video communication applications, we established the d-R-D model for IPPPP coding mode suitable in real-time applications, and d-R-D model for hierarchical B pictures utilized in delay insensitive applications respectively. We adopt the distortion model from Ref [89] and rate model from Ref [90] as R-D model, where R-D are functions of prediction residual variance and quantization parameters.

Next, we relate the residual variance with tunable delay parameters, ME complexity parameters for IPPPP mode and GOP size for hierarchical B pictures specifically, so as to write R-D as functions of the delay parameters. We successfully developed d-R-D model for the two coding modes based on this idea. Experiment results show that the proposed d-R-D models are quite accurate when the method of compensation from neighboring frame/GOP is applied to estimate coding rate and distortion. And in both coding modes, we apply the proposed d-R-D model to rate distortion optimization based rate control. In IPPPP mode, we conduct RDO on both sequence level and frame level rate control by finding the optimal search range and quantization parameter for the first time. The derived search range can be seen as a motion estimation threshold, over which the motion estimation will contribute very little to coding efficiency, or simply put, any motion estimation out of the threshold search bound will be a waste of computation power. In hierarchical B picture, the RDO rate control obtains the optimal GOP size and quantization parameter corresponding to that GOP size. Although it has been reported in many literature that the structure delay parameter N_{gop} can affect the coding efficiency, no mathematical analysis ever appears to specify how to find the optimal N_{gop} in rate distortion sense. With the proposed d-R-D model for hierarchical B picture, we developed an applicable algorithm to derive the best GOP size based on rate distortion optimization rate control for the first time.

Furthermore, to verify the source coding d-R-D model for IPPPP mode is applicable in delay sensitive wireless applications, we conduct extensive simulations of video communication over wireless channels, and obtain the transmission performance with varying encoding time tunable in d-R-D model. With this model, the encoding time can be adaptively adjusted to vary the transmission delay which will influence the transmission system performance. The overall system performance is a combination of both source coding system and transmission system. Hence, by tuning the encoding time via d-R-D model, we can optimize the overall system performance.

Note that when we do the RDO rate control in hierarchical B picture, our target is to minimize the average distortion over a group of picture, and the optimal GOP size is ”optimal“ in an average sense of the entire sequence, not optimal for every single GOP. Adaptive GOP structure (AGS) has been proposed in scalable video coding to further improve the coding efficiency [141] [142][143]. After core experiment (CE) 2 that shows coding efficiency up to 0.62dB improve [144], it has been adopted in SVC [131][145][146][147]. The concept of AGS is to adaptively select GOP size by considering the temporal image characteristics of the video sequence, while the implementation is to exhaustively search over all possible coding structure variation of a group of pictures, e.g. 16 frames full-GOP. Some fast algorithms later appear to reduce the computation complexity of AGS [148]. The proposed d-R-D model can also achieve AGS by updating the model parameters on a temporal basis, and apply RDO to find the best GOP size for that time period. This would be one of our future work to expand the application of the d-R-D model. In addition, current d-R-D model of hierarchical B picture does not take into account the coding problem of layer dependency, i.e. frames on layer k use frames on layer $k-1$ for reference. We may refer to works that addressed the coding dependency to improve the model accuracy without compensation from the neighboring GOP [140][149][150][151][152][140].

CHAPTER 6
CONCLUSION

In this work, we addressed four problems in a typical video communication system: image and video denoising in pre-processing, compression efficiency in video coding, video content protection and delay constraint in video streaming.

In Chapter 2, we proposes a bounded BM3D scheme. The novelty of our bounded BM3D is two-folded. First, our scheme partitions an image into multiple regions, and identifies the boundaries between regions. And we restrict block matching search within the region of the template block. Second, to prevent important geometric features such as edges from being removed by collaborative filtering in BM3D, we do partial block matching for different block coherent segments which belong to different regions. Compared to BM3D, the proposed bounded BM3D achieves better visual performance, as well as 0.23dB-1.33dB PSNR increase for heavily noisy images.

In Chapter 3, we propose a classified quadtree-based adaptive loop filter (CQALF) in video coding. Pixels in a picture are classified into two categories by considering the impact of the deblocking filter, the pixels that are modified and the pixels that are not modified by the deblocking filter. A wiener filter is carefully designed for each category and the filter coefficients are transmitted to decoder. For the pixels that are modified by the deblocking filter, the filter is estimated at encoder by minimizing the mean square error between the original input frame and a combined frame which is a weighted average of the reconstructed frames before and after the deblocking filter. For pixels that the deblocking filter does not modify, the filter is estimated by minimizing the mean square error between the original frame and the reconstructed frame. The proposed algorithm is implemented on top of KTA software and compatible with the quadtree-based adaptive loop filter. Compared with kta2.6r1 anchor, the proposed CQALF achieves 10.05%, 7.55%, and 6.19% BD bitrate reduction in average for intra only, IPPP, and HB coding structures respectively.

In Chapter 4, we propose a video hash scheme that utilizes image hash and spatio-temporal information contained in video to generate video hash. A video clip is firstly segmented to shots, and video hash is derived in unit of shot. We notice that for video hash applications in identification and verification, reference video and suspected video always appear in pair. Therefore, we propose to derive the shot hash in a pairwise manner. For both reference and suspected shot, we apply 2-Dimensional Discrete Cosine Transform (2D-DCT) to each frame in the shot, quantize the Discrete Cosine Transform (DCT) coefficient, and record the temporal occurrence of the co-located coefficient. We then choose a pair of the closet value as the DCT coefficient for every collocated entry, inverse transform to the spatial domain, and derive image hashes from two feature frames by hash based on Radial projections (Radial hASH). Experiment results show that the proposed 2D-DCT temporal maximum occurrence (2D-DCT TMO) scheme successfully derives shot hash that represents the content, and is very robust in video identification, authentication, and verification.

In Chapter 5, we propose a source coding delay-Rate-Distortion model for the first time to study the relationship among the three coding variants. Due to the different delay constraint in real-time and none real-time video communication applications, we established the d-R-D model for IPPPP coding mode suitable in real-time applications, and d-R-D model for hierarchical B pictures utilized in delay insensitive applications respectively. We adopt the distortion model from Ref [89] and rate model from Ref [90] as R-D model, where R-D are functions of prediction residual variance and quantization parameters. Next, we relate the residual variance with tunable delay parameters, ME complexity parameters for IPPPP mode and GOP size for hierarchical B pictures specifically, so as to write R-D as functions of the delay parameters. We successfully developed d-R-D model for the two coding mode based on this idea. Experiment results show that the proposed d-R-D models are quite accurate when the method of compensation from neighboring frame/GOP is applied to estimate coding rate and

distortion. Furthermore, to verify the source coding d-R-D model for IPPPP mode is applicable in delay sensitive wireless applications, we conduct extensive simulations of video communication over wireless channels, and obtain the transmission performance with varying encoding time tunable in d-R-D model.

REFERENCES

[1] A. Buades, B. Coll, and J. Morel, “A review of image denoising algorithms, with a new one,” *Multiscale Modeling and Simulation*, vol. 4, no. 2, pp. 490–530, 2006.

[2] L. Yaroslavsky, K. Egiazarian, and J. Astola, “Transform domain image restoration methods: review, comparison, and interpretation,” in *Proceedings of SPIE*, vol. 4304, 2001, p. 155.

[3] S. Karunasekera and N. Kingsbury, “A distortion measure for blocking artifacts in images based onhuman visual sensitivity,” *IEEE Transactions on image processing*, vol. 4, no. 6, pp. 713–724, 1995.

[4] H. Reeve III and J. Lim, “Reduction of blocking effect in image coding,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing, 1983.*, vol. 8, 1983.

[5] A. Netravali and B. Haskell, *Digital pictures: representation, compression, and standards*. Plenum Pub Corp, 1995.

[6] R. Plompen and R. Plompen, *Motion video coding for visual telephony*. PTT Research Neher Laboratories, 1989.

[7] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising by sparse 3-D transform-domain collaborative filtering,” *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[8] L. Sendur and I. Selesnick, “Bivariate shrinkage functions for wavelet-based denoising exploiting interscale dependency,” *IEEE Transactions on Signal Processing*, vol. 50, no. 11, pp. 2744–2756, 2002.

[9] A. Pizurica, W. Philips, I. Lemahieu, and M. Acheroy, “A joint inter-and intrascale statistical model for Bayesian wavelet based image denoising,” *IEEE Transactions on Image Processing*, vol. 11, no. 5, pp. 545–557, 2002.

[10] J. Portilla, V. Strela, M. Wainwright, and E. Simoncelli, “Image denoising using scale mixtures of Gaussians in the wavelet domain,” *IEEE Transactions on Image Processing*, vol. 12, no. 11, pp. 1338–1351, 2003.

[11] J. Guerrero-Colon and J. Portilla, “Two-level adaptive denoising using Gaussian scale mixtures in overcomplete oriented pyramids,” in *IEEE International Conference on Image Processing*, vol. 1. IEEE, 2005.

[12] D. Muresan and T. Parks, “Adaptive principal components and image denoising,” in *International Conference on Image Processing, 2003.*, vol. 1. IEEE, 2003.

[13] I. Richardson, *H.264 and MPEG-4 video compression*. Wiley Online Library, 2003.

[14] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, “Overview of the H. 264/AVC video coding standard,” *IEEE Transactions on circuits and systems for video technology*, vol. 13, no. 7, pp. 560–576, 2003.

[15] Y. Liu, “Unified Loop Filter for Video Compression,” *IEEE Transactions on Circuits and Systems for Video Technology*, no. 99, p. 1, 2010.

[16] I. Cox, M. Miller, K. Tanaka, and Y. Wakasu, “Digital watermarking,” Jun. 22 1999, uS Patent 5,915,027.

[17] M. Wu and B. Liu, *Multimedia data hiding*. Springer Verlag, 2003.

[18] C. De Roover, C. De Vleeschouwer, F. Lefebvre, and B. Macq, “Robust video hashing based on radial projections of key frames,” *IEEE Transactions on Signal processing*, vol. 53, no. 10, pp. 4020–4037, 2005.

[19] A. Mucedero, R. Lancini, and F. Mapelli, “A novel hashing algorithm for video sequences,” in *Proceedings of the 2004 International Conference on Image Processing*, vol. 4, 2004, pp. 2239–2242.

[20] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising with block-matching and 3 d filtering,” in *Proceedings of SPIE*, vol. 6064. Citeseer, 2006, pp. 354–365.

[21] E. Vansteenkiste, D. Van der Weken, W. Philips, and E. Kerre, “Perceived image quality measurement of state-of-the-art noise reduction schemes,” in *Advanced Concepts for Intelligent Vision Systems*. Springer, 2006, pp. 114–126.

[22] A. Horé and D. Ziou, “Image quality metrics: Psnr vs. ssim,” in *International Conference on Pattern Recognition*. IEEE, 2010, pp. 2366–2369.

[23] T. Sikora, “Low complexity shape-adaptive DCT for coding of arbitrarily shaped image segments,” *Signal processing: Image communication*, vol. 7, no. 4-6, pp. 381–395, 1995.

[24] A. Foi, V. Katkovnik, and K. Egiazarian, “Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images,” *IEEE Transactions on Image Processing*, vol. 16, no. 5, p. 1395, 2007.

[25] M. Yuen and H. Wu, “A survey of hybrid MC/DPCM/DCT video coding distortions,” *Signal Processing*, vol. 70, no. 3, pp. 247–278, 1998.

[26] “Bm3d software,” *[Online]*.

[27] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Color image denoising via sparse 3d collaborative filtering with grouping constraint in luminance-chrominance space,” in *IEEE International Conference on Image Processing, 2007.*, vol. 1. IEEE, 2007, pp. I–313.

[28] A. Foi, V. Katkovnik, and K. Egiazarian, “Pointwise shape-adaptive dct denoising with structure preservation in luminance-chrominance space,” in *Proceedings of the 2nd International Workshop on Video Processing and Quality Metrics for Consumer Electronics*. Citeseer, 2006.

[29] ——, “Pointwise shape-adaptive dct for high-quality denoising and deblocking of grayscale and color images,” *IEEE Transactions on Image Processing*, vol. 16, no. 5, pp. 1395–1411, 2007.

[30] K. Dabov, A. Foi, and K. Egiazarian, “Video denoising by sparse 3d transform-domain collaborative filtering,” in *Proceedings of 15th European Signal Processing Conference*, vol. 1, no. 2. Citeseer, 2007, p. 7.

[31] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Joint image sharpening and denoising by 3d transform-domain collaborative filtering,,” in *Proceedings of Int. TICSP Workshop Spectral Meth. Multirate Signal Process*, 2007.

[32] M. Mäkitalo and A. Foi, “Spatially adaptive alpha-rooting in bm3d sharpening,” in *Society of Photo-Optical Instrumentation Engineers Conference Series*, vol. 7870, 2011, p. 32.

[33] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image restoration by sparse 3d transform-domain collaborative filtering,” in *SPIE Electronic Imaging*. Citeseer, 2008.

[34] A. Danielyan, V. Katkovnik, and K. Egiazarian, “Image deblurring by augmented langrangian with bm3d frame prior,” in *Workshop on Information Theoretic Methods in Science and Engineering*, 2010.

[35] ——, “Bm3d frames and variational image deblurring,” *to appear in IEEE Transactions on Image Processing*, 2011.

[36] A. Danielyan, A. Foi, V. Katkovnik, and K. Egiazarian, “Image upsampling via spatially adaptive block-matching filtering ,” in *Proceedings of 16th European Signal Processing Conference, EUSIPCO2008*. Citeseer, 2008.

[37] ——, “Image and video super-resolution via spatially adaptive block-matching filtering,” in *Proceedings of International Workshop on Local and Non-Local Approximation in Image Processing*. Citeseer, 2008.

[38] G. Boracchi and A. Foi, “Multiframe raw-data denoising based on block-matching and 3-d filtering for low-light imaging and stabilization,” in *Proceedings of International Workshop on Local and Non-Local Approximation in Image Processing*, 2008.

[39] A. Danielyan, M. Vehvilainen, A. Foi, V. Katkovnik, and K. Egiazarian, “Cross-color bm3d filtering of noisy raw data,” in *Proceedings of International Workshop on*

Local and Non-Local Approximation in Image Processing. IEEE, 2009, pp. 125–129.

[40] M. Maggioni, G. Boracchi, A. Foi, and K. Egiazarian, "Video denoising using separable 4-d nonlocal spatiotemporal transforms," in *Society of Photo-Optical Instrumentation Engineers Conference Series*, vol. 7870, 2011, p. 1.

[41] S. Abramov, V. Lukin, B. Vozel, K. Chehdi, and J. Astola, "Segmentation-based method for blind evaluation of noise variance in images," *Journal of Applied Remote Sensing*, vol. 2, 2008.

[42] D. Donoho and J. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, no. 3, p. 425, 1994.

[43] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian, "Practical poissonian-gaussian noise modeling and fitting for single-image raw-data," *IEEE Transactions on Image Processing*, vol. 17, no. 10, pp. 1737–1754, 2008.

[44] "The influence curve and its role in robust estimation," *Journal of American Statistical Association*, pp. 1179–1186, 1974.

[45] N. Ponomarenko, V. Lukin, M. Zriakhov, A. Kaarna, and J. Astola, "An automatic approach to lossy compression of aviris images," in *IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2007, pp. 472–475.

[46] A. Danielyan and A. Foi, "Noise variance estimation in nonlocal transform domain," in *Proceedings of International Workshop on Local and Non-Local Approximation in Image Processing*. IEEE, 2009, pp. 41–45.

[47] S. Wittmann and T. Wedi, "Transmission of post-filter hints for video coding schemes," in *IEEE International Conference on Image Processing, 2007.*, vol. 1. IEEE, 2007.

[48] T. A. Chujoh, T. and T. Yamakage, "Adaptive Loop Filter for Improving Coding Efficiency," in *ITU-T SG16 Contribution, C402, Geneva, April*, 2008.

[49] Y. Chiu and L. Xu, "Adaptive (Wiener) Filter for Video Compression," in *ITU-T SG16 Contribution, C437, Geneva, April*, 2008.

[50] T. A. Chujoh, T. and T. Yamakage, "Block-based Adaptive loop Filter," in *ITU-T SG16 Q.6 Document, VCEG-AJ13, San Diego, October*, 2008.

[51] T. Watanabe, N. Wada, G. Yasuda, A. Tanizawa, T. Chujoh, and T. Yamakage, "In-loop filter using block-based filter control for video coding," in *IEEE International Conference on Image Processing*. IEEE, 2009, pp. 1013–1016.

[52] H. Jo, D. Sim, and H. Kim, "Macroblock-based adaptive loop filter for video compression," in *International Conference on Multimedia and Signal Processing*, vol. 2. IEEE, 2011, pp. 366–369.

[53] W. N. W. T. Chujoh, T. and G. Yasuda, “Specification and Experimental Results of Quadtree-based Adaptive Loop Filter,” in *ITU-T SG16, Q.6 Document, VCEG-AK22, Japan, April*, 2009.

[54] C. P. Karczewicz, M. and Rajan, “Video coding technology proposal by Qualcomm Inc.” in *ITU-T SG16, JCTVC-A121, Germany, April*, 2010.

[55] T. T. W. S. F. E. Jung, J., “Description of video coding technology proposal by France Telecom, NTT, NTT DOCOMO, Panasonic and Technicolor,” in *ITU-T SG16, JCTVC-A114, German, April*, 2010.

[56] G. Raja and M. Mirza, “In-loop deblocking filter for h. 264/avc video,” *Proceedings of ISCCSP*, 2006.

[57] P. List, A. Joch, J. Lainema, G. Bjontegaard, and M. Karczewicz, “Adaptive deblocking filter,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 614–619, 2003.

[58] S. Shih, C. Chang, and Y. Lin, “A near optimal deblocking filter for h. 264 advanced video coding,” in *Proceedings of the 2006 Asia and South Pacific Design Automation Conference*. IEEE Press, 2006, pp. 170–175.

[59] Y. Zhong, I. Richardson, A. Miller, and Y. Zhao, “Perceptual quality of h.264/avc deblocking filter,” in *IEE CONFERENCE PUBLICATION*, vol. 509. Citeseer, 2005, p. 379.

[60] T. Chujoh, G. Yasuda, N. Wada, T. Watanabe, and T. Yamakage, “Improvement of block-based adaptive loop filter,” *ITU-T SG16/Q*, vol. 6.

[61] T. Chujoh, N. Wada, and G. Yasuda, “Quadtree-based adaptive loop filter,” *ITU-T SG16 Contribution, C*, vol. 181.

[62] T. Chujoh, T. Yamakage, and G. Yasuda, “Improvement of quadtree-based adaptive loop filter,” *ITU-T SG16 Contribution, C*, vol. 324.

[63] H. Li, Y. Li, C. Wu, R. Song, and Y. Feng, “Fast multi-symmetry adaptive loop filter algorithm,” in *Proceedings of SPIE*, vol. 7810, 2010, p. 78100Y.

[64] W. Yang, “Image/video processing techniques: header compression and post-processing,” 2010.

[65] T. Tan, G. Sullivan, and T. Wedi, “Recommended simulation common conditions for coding efficiency experiments revision 1,” *ITU-T Q*, vol. 6, pp. 8–10, 2008.

[66] S. Pateux and J. Jung, “An excel add-in for computing Bjontegaard metric and its evolution,” in *VCEG document VCEG-AE07, 31st VCEG Meeting, Marrakech, MA*, 2007, pp. 15–16.

[67] Z. Y. L. X. Chen, Q. and P. Yin, “Adaptive loop filtering,” *PU100244*, 2011.

[68] B. Coskun, B. Sankur, and N. Memon, “Spatio–Temporal Transform Based Video Hashing,” *IEEE Transactions on Multimedia*, vol. 8, no. 6, pp. 1190–1208, 2006.

[69] J. Oostveen, T. Kalker, and J. Haitsma, “Visual hashing of digital video: applications and techniques,” in *Proceedings of SPIE*, vol. 4472, 2001, p. 121.

[70] X. Zhou, M. Schmucker, and C. Brown, “Perceptual hashing of video content based on differential block similarity,” *Computational Intelligence and Security 2005 Proceedings, Part II: LNAI 3802*, pp. 80–85, 2005.

[71] R. Venkatesan, S. Koon, M. Jakubowski, and P. Moulin, “Robust image hashing,” in *IEEE International Conference on Image Processing*, vol. 3, 2000.

[72] J. Fridrich and M. Goljan, “Robust hash functions for digital watermarking,” in *International Conference on Information Technology: Coding and Computing*, 2000, pp. 178–183.

[73] M. Mıhçak and R. Venkatesan, “New Iterative Geometric Methods for Robust Perceptual Image Hashing,” in *Security and privacy in digital rights management*. Springer Verlag, 2002, pp. 13–21.

[74] A. Swaminathan, Y. Mao, and M. Wu, “Robust and secure image hashing,” *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 2, pp. 215–230, 2006.

[75] J. Boreczky and L. Rowe, “Comparison of video shot boundary detection techniques,” *Journal of Electronic Imaging*, vol. 5, no. 2, pp. 122–128, 1996.

[76] C. Cai, K. Lam, and Z. Tan, “An efficient video shot representation for fast video retrieval,” in *Proceedings of SPIE Visual Communications and Image Processing*, vol. 5960, 2005.

[77] X. Dong, X. Li, and D. Wu, “Analysis of packet error probability in delay constrained communication over fading channels,” in *Consumer Communications and Networking Conference, 2009.* IEEE, 2009, pp. 1–5.

[78] C. Hsu and A. Ortega, “A Lagrangian optimization approach to rate control for delay-constrained video transmission over burst-error channels,” in *IEEE International Conference on Acoustics, Speech and Signal Processing, 1998.*, vol. 5. IEEE, 1998, pp. 2989–2992.

[79] C. Hsu, A. Ortega, and M. Khansari, “Rate control for robust video transmission over burst-error wireless channels,” *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 5, pp. 756–773, 1999.

[80] T. Wang, H. Fang, and L. Chen, “Low delay, error robust wireless video transmission architecture for video communication,” in *IEEE International Conference on Multimedia and Expo, 2002.*, vol. 1. IEEE, 2002, pp. 265–268.

[81] Q. Chen and D. Wu, “Delay constrained video transmission over wireless channels,” *IEEE COMSOC MMTC E-letter*, vol. 5, no. 1, 2010.

[82] M. van der Schaar and D. Turaga, “Cross-layer packetization and retransmission strategies for delay-sensitive wireless multimedia transmission,” *IEEE Transactions on Multimedia*, vol. 9, no. 1, pp. 185–197, 2007.

[83] Z. Y. J. R. Liang Zhou, Min Chen and H.-C. Chao, “Cross-layer wireless video adaptation: Tradeoff between distortion and delay,” *Computer Communications*, pp. 1615–1622, 2010.

[84] X. Dong, X. Li, and D. Wu, “Theory for QoS Provisioning in Wireless Networks,” *submitted to IEEE Transactions on Wireless Communications*, 2010.

[85] [online], “Delay in video transmission and ratecontrol,” 2005.

[86] C. Hsu, A. Ortega, and A. Reibman, “Joint selection of source and channel rate for VBR video transmission under ATM policing constraints,” *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 6, pp. 1016–1028, 1997.

[87] D. Turaga, M. van der Schaar, and B. Pesquet-Popescu, “Complexity scalable motion compensated wavelet video encoding,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 8, pp. 982–993, 2005.

[88] Z. He, Y. Liang, L. Chen, I. Ahmad, and D. Wu, “Power-rate-distortion analysis for wireless video communication under energy constraints,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 5, pp. 645–658, 2005.

[89] X. Li, N. Oertel, A. Hutter, and A. Kaup, “Laplace distribution based Lagrangian rate distortion optimization for hybrid video coding,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 2, pp. 193–205, 2009.

[90] Z. Chen and D. Wu, “Rate-distortion optimized cross-layer rate control in wireless video communication,” *IEEE Transactions on Circuits and Systems for Video Technology*, 2011.

[91] J. Lorch and A. Smith, “Improving dynamic voltage scaling algorithms with pace,” in *ACM SIGMETRICS Performance Evaluation Review*, vol. 29, no. 1. ACM, 2001, pp. 50–61.

[92] R. Min, T. Furrer, and A. Chandrakasan, “Dynamic voltage scaling techniques for distributed microsensor networks,” in *IEEE Computer Society Workshop on VLSI Proceedings.* IEEE, 2000, pp. 43–46.

[93] T. Cover, J. Thomas, and MyiLibrary, *Elements of information theory*. Wiley Online Library, 1991, vol. 6.

[94] Z. He and S. Mitra, "Optimum bit allocation and accurate rate control for video coding via ρ-domain source modeling," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 10, pp. 840–849, 2002.

[95] T. Chiang and Y. Zhang, "A new rate control scheme using quadratic rate distortion model," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 1, pp. 246–250, 1997.

[96] J. Ribas-Corbera and S. Lei, "Rate control in dct video coding for low-delay communications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 1, pp. 172–185, 1999.

[97] N. Kamaci, Y. Altunbasak, and R. Mersereau, "Frame bit allocation for the h. 264/avc video coder via cauchy-density-based rate and distortion models," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 8, pp. 994–1006, 2005.

[98] Z. He and S. Mitra, "A linear source model and a unified rate control algorithm for dct video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 11, pp. 970–982, 2002.

[99] S. Ma, W. Gao, and Y. Lu, "Rate-distortion analysis for h. 264/avc video coding and its application to rate control," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 12, pp. 1533–1544, 2005.

[100] A. Leontaris and P. Cosman, "Compression efficiency and delay tradeoffs for hierarchical b-pictures and pulsed-quality frames," *IEEE Transactions on Image Processing*, vol. 16, no. 7, pp. 1726–1740, 2007.

[101] F. Moscheni, F. Dufaux, and H. Nicolas, "Entropy criterion for optimal bit allocation between motion and prediction error information," in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, vol. 2094. Citeseer, 1993, pp. 235–242.

[102] I.-T. S. H:, "Audiovisual and multimedia systems, advanced video coding for generic audiovisual services," Nov. 2007.

[103] "H.264/avc reference software jm16.0," *[Online]*.

[104] "Videolan," *[Online]*.

[105] L. Merritt and R. Vanam, "x264: A high performance h. 264/avc encoder," *[online]*.

[106] H. Everett III, "Generalized lagrange multiplier method for solving problems of optimum allocation of resources," *Operations Research*, pp. 399–417, 1963.

[107] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers [speech coding]," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 36, no. 9, pp. 1445–1453, 1988.

[108] K. Ramchandran and M. Vetterli, “Best wavelet packet bases in a rate-distortion sense,” *Image Processing, IEEE Transactions on*, vol. 2, no. 2, pp. 160–175, 1993.

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

[110] I. Richardson, *The H. 264 advanced video compression standard*. Wiley, 2010.

[111] A. Ortega and K. Ramchandran, “Rate-distortion methods for image and video compression,” *Signal Processing Magazine, IEEE*, vol. 15, no. 6, pp. 23–50, 1998.

[112] Z. Chen and D. Wu, “Prediction of transmission distortion for wireless video communication: Algorithm and application,” *Journal of Visual Communication and Image Representation*, 2010.

[113] A. Goldsmith, *Wireless communications*. Cambridge University Press, 2005.

[114] J. Kwon and J. Kim, “Adaptive code rate decision of joint source-channel coding for wireless video,” *Electronics Letters*, vol. 38, no. 25, pp. 1752–1754, 2002.

[115] S. Lin and D. Costello, *Error control coding: fundamentals and applications*. Prentice-hall Englewood Cliffs, NJ, 1983.

[116] P. van Beek and M. Demircin, “Delay-constrained rate adaptation for robust video transmission over home networks,” in *IEEE International Conference on Image Processing*, vol. 2. IEEE, 2005, pp. II–173.

[117] S. K. Jeong, Y. and P. K.H, “Analysis of packet transmission delay for adaptation of fec code rate and packet size in real-time joint source-channel coding,” in *draft*.

[118] A. Hoang and M. Motani, “Cross-layer adaptive transmission: Optimal strategies in fading channels,” *IEEE Transactions on Communications*, vol. 56, no. 5, pp. 799–807, 2008.

[119] R. Berry and R. Gallager, “Communication over fading channels with delay constraints,” *IEEE Transactions on Information Theory*, vol. 48, no. 5, pp. 1135–1149, 2002.

[120] A. Fu, E. Modiano, and J. Tsitsiklis, “Optimal energy allocation for delay-constrained data transmission over a time-varying channel,” in *INFOCOM 2003*, vol. 2. IEEE, 2003, pp. 1095–1105.

[121] K. Stuhlmuller, N. Farber, M. Link, and B. Girod, “Analysis of video transmission over lossy channels,” *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 6, pp. 1012–1032, 2000.

[122] H. Schwarz, D. Marpe, and T. Wiegand, “Hierarchical b pictures,” *Joint Video Team, Doc. JVT-P014, Poznan, Poland*, 2005.

[123] B. Girod, “The efficiency of motion-compensating prediction for hybrid coding of video sequences,” *IEEE Journal on Selected Areas in Communications*, vol. 5, no. 7, pp. 1140–1154, 1987.

[124] ——, “Efficiency analysis of multihypothesis motion-compensated prediction for video coding,” *IEEE Transactions on Image Processing*, vol. 9, no. 2, pp. 173–183, 2000.

[125] Y. Cho, C. Kuo, and D. Kwon, “Gop-based rate control for h.264/svc with hierarchical b-pictures,” *iih-msp*, pp. 387–390, 2007.

[126] Z. Li, F. Pan, K. Lim, G. Feng, X. Lin, and S. Rahardja, “Adaptive basic unit layer rate control for jvt,” in *JVT-G012-r1, 7th Meeting, Pattaya II, Thailand*, 2003.

[127] B. Y. Karczewicz. M, “Need for further avc test model enhancement,” in *JVT-L034, Redmond*, 2004.

[128] G. Pau, J. Viéron, and B. Pesquet-Popescu, “Video coding with flexible mctf structures for low end-to-end delay,” in *IEEE International Conference on Image Processing, 2005.*, vol. 3. IEEE, 2005, pp. III–241.

[129] G. Pau, B. Pesquet-Popescu, M. van der Schaar, J. Vieron, and J. Viéron, “Delay-performance trade-offs in motion-compensated scalable subband video compression,” in *Proceedings of Advanced Concepts for Intelligent Vision Systems (ACIVS)*. Citeseer, 2004.

[130] H. Schwarz, D. Marpe, and T. Wiegand, “Overview of the scalable h. 264/mpeg4-avc extension,” in *Image Processing, 2006 IEEE International Conference on*. IEEE, pp. 161–164.

[131] “Coding of audio-visual objectsłpart 10: Advanced video coding; amendment 3 scalable video coding,” in *ISO/IEC 14496-1*, 2005.

[132] M. Wien, H. Schwarz, and T. Oelbaum, “Performance analysis of svc,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 17, no. 9, pp. 1194–1203, 2007.

[133] H. Schwarz, D. Marpe, and T. Wiegand, “Analysis of hierarchical b pictures and mctf,” in *IEEE International Conference on Multimedia and Expo*. IEEE, 2006, pp. 1929–1932.

[134] ——, “Mctf and scalability extension of h. 264/avc,” in *Proceedings of PCS*. Citeseer, 2004.

[135] R. Schäfer, H. Schwarz, D. Marpe, T. Schierl, and T. Wiegand, “Mctf and scalability extension of h. 264/avc and its application to video transmission, storage, and surveillance,” in *Proceedings of SPIE*, vol. 5960, 2005, pp. 343–354.

[136] H. Schwarz, D. Marpe, and T. Wiegand, “Comparison of mctf and closed-loop hierarchical b pictures,” in *JVT-P059, Poznan, Poland*, 2005.

[137] A. Leontaris and P. Cosman, “End-to-end delay for hierarchical b-pictures and pulsed quality dual frame video coders,” in *IEEE International Conference on Image Processing, 2006*. IEEE, 2006, pp. 3133–3136.

[138] K. Lim, G. Sullivan, and T. Wiegand, “Text description of joint model reference encoding methods and decoding concealment methods,” *JVT of ISO/IEC MPEG and ITU-T VCEG, JVT-K*, 2004.

[139] S. Ma, Z. Li, and F. Wu, “Proposed draft of adaptive rate control,” in *Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, Doc. JVT-H017r3, 8th Meeting, Geneva*, 2003, pp. 20–26.

[140] K. Ramchandran, A. Ortega, and M. Vetterli, “Bit allocation for dependent quantization with applications to multiresolution and mpeg video coders,” *IEEE Transactions on Image Processing*, vol. 3, no. 5, pp. 533–545, 1994.

[141] G. Park, M. Park, S. Jeong, J. Cha, K. Kim, and J. Hong, “Adaptive gop structure for svc,” *ISO/IEC JTC1/SC29/WG11, M*, vol. 11563, p. 2005.

[142] G. Park, M. Park, S. Jeong, K. Kim, and J. Hong, “Improve svc coding efficiency by adaptive gop structure (svc ce2),” *ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q*, vol. 6.

[143] Y. Wang and F. Liang, “Improved adaptive group of pictures structure,” *Electronics Letters*, vol. 42, no. 21, pp. 1210–1211, 2006.

[144] S. Jeannin and M. Bober, “Description of core experiments for mpeg-7 motion/shape,” *MPEG-7, ISO/IEC/JTC1/SC29/WG11/MPEG99*, vol. 2690, p. 1999.

[145] J. Reichel, H. Schwarz, and M. Wien, “Joint scalable video model 2.0 reference encoding algorithm description,” *ISO/IEC JTC1/SC29/WG11*, vol. 7556.

[146] J. Model, “Jsvm software, joint video team, doc,” *JVT-X203, Geneva, Switzerland*, 2007.

[147] J. Reichel, H. Schwarz, and M. Wien, “Joint scalable video model 11 (jsvm 11),” *Joint Video Team, Doc. JVT- X*, 2007.

[148] Y. Chen, C. Lin, C. Chen, and L. Chen, “Fast prediction algorithm of adaptive gop structure for svc,” in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, vol. 6508, 2007, p. 29.

[149] P. Burt and E. Adelson, “The laplacian pyramid as a compact image code,” *IEEE Transactions on Communications*, vol. 31, no. 4, pp. 532–540, 1983.

[150] K. Uz, M. Vetterli, and D. LeGall, “Interpolative multiresolution coding of advance television with compatible subchannels,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 1, no. 1, pp. 86–99, 1991.

[151] D. Gall, “Mpeg: A video compression standard for multimedia applications,” *ACM Communication.*, vol. 34, no. 4, pp. 46–58, 1991.

[152] K. Ramchandran, A. Ortega, K. Uz, and M. Vetterli, “Multiresolution broadcast for digital hdtv using joint source/channel coding,” *IEEE Journal on Selected Areas in Communications*, vol. 11, no. 1, pp. 6–23, 1993.

BIOGRAPHICAL SKETCH

Qian Chen received the B.E. degree in Telecommunications from Xidian University, Xi'an, China in 2005, and the M.S. in Electronic and Electrical Engineering from Shanghai Jiao Tong University, Shanghai, China in 2008. Currently she is a PhD candidate in Electrical and Computer Engineering with University of Florida.

From May 2010 to Aug 2010, she was an intern in Technicolor, Princeton, NJ, where she worked in video coding for next generation video compression (HEVC). From May 2011 to Aug. 2011, she worked in InterDigital Communications LLC as a research intern on HEVC video coding standard and algorithm testing. In Sep. 2011,she joined Dolby Laboratories Inc. as Senior Design Engineer, conducting research and implementation on Dolby's next generation video technology.

Her research interests include video coding and transmission, multimedia security, image and video processing. She has published several journal and conference papers, and filed 5 patents. She is also the reviewer for IEEE Transaction on Circuits System and Video Technology, Journal of Visual Communication & Image Representation, Journal of Signal Processing.