Image and Video Coding/Transcoding: A Rate Distortion Approach

by

Xiang Yu

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2008

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Due to the lossy nature of image/video compression and the expensive bandwidth and computation resources in a multimedia system, one of the key design issues for image and video coding/transcoding is to optimize trade-off among distortion, rate, and/or complexity. This thesis studies the application of rate distortion (RD) optimization approaches to image and video coding/transcoding for exploring the best RD performance of a video codec compatible to the newest video coding standard H.264 and for designing computationally efficient down-sampling algorithms with high visual fidelity in the discrete Cosine transform (DCT) domain.

RD optimization for video coding in this thesis considers two objectives, i.e., to achieve the best encoding efficiency in terms of minimizing the actual RD cost and to maintain decoding compatibility with the newest video coding standard H.264. By the actual RD cost, we mean a cost based on the final reconstruction error and the entire coding rate. Specifically, an operational RD method is proposed based on a soft decision quantization (SDQ) mechanism, which has its root in a fundamental RD theoretic study on fixed-slope lossy data compression. Using SDQ instead of hard decision quantization, we establish a general framework in which motion prediction, quantization, and entropy coding in a hybrid video coding scheme such as H.264 are jointly designed to minimize the actual RD cost on a frame basis. The proposed framework is applicable to optimize any hybrid video coding scheme, provided that specific algorithms are designed corresponding to coding syntaxes of a given standard codec, so as to maintain compatibility with the standard.

Corresponding to the baseline profile syntaxes and the main profile syntaxes of H.264, respectively, we have proposed three RD algorithms—a graph-based algorithm for SDQ given motion prediction and quantization step sizes, an algorithm for residual coding optimization given motion prediction, and an iterative overall algorithm for jointly optimizing motion prediction, quantization, and entropy coding—with them embedded in the indicated order. Among the three algorithms, the SDQ design is the core, which is developed based on a given entropy coding

method. Specifically, two SDQ algorithms have been developed based on the context adaptive variable length coding (CAVLC) in H.264 baseline profile and the context adaptive binary arithmetic coding (CABAC) in H.264 main profile, respectively.

Experimental results for the H.264 baseline codec optimization show that for a set of typical testing sequences, the proposed RD method for H.264 baseline coding achieves a better trade-off between rate and distortion, i.e., 12% rate reduction on average at the same distortion (ranging from 30dB to 38dB by PSNR) when compared with the RD optimization method implemented in H.264 baseline reference codec. Experimental results for optimizing H.264 main profile coding with CABAC show 10% rate reduction over a main profile reference codec using CABAC, which also suggests 20% rate reduction over the RD optimization method implemented in H.264 baseline reference codec, leading to our claim of having developed the best codec in terms of RD performance, while maintaining the compatibility with H.264.

By investigating trade-off between distortion and complexity, we have also proposed a designing framework for image/video transcoding with spatial resolution reduction, i.e., to down-sample compressed images/video with an arbitrary ratio in the DCT domain. First, we derive a set of DCT-domain down-sampling methods, which can be represented by a linear transform with double-sided matrix multiplication (LTDS) in the DCT domain. Then, for a pre-selected pixel-domain down-sampling method, we formulate an optimization problem for finding an LTDS to approximate the given pixel-domain method to achieve the best trade-off between visual quality and computational complexity. The problem is then solved by modeling an LTDS with a multi-layer perceptron network and using a structural learning with forgetting algorithm for training the network. Finally, by selecting a pixel-domain reference method with the popular Butterworth lowpass filtering and cubic B-spline interpolation, the proposed framework discovers an LTDS with better visual quality and lower computational complexity when compared with state-of-the-art methods in the literature.

Acknowledgements

There are many people I wish to thank for their support and contributions during the course of my Ph.D. First of all, I wish to express my sincere gratitude to my advisor, Professor En-hui Yang for his insightful guidance, great patience, and generous support. I have benefited greatly from the extensive research training he has provided me and from his wisdom and exceptional knowledge of the field. His cutting-edge researches in abstract theory and his achievements with innovations that impact the real world have always been a great source of inspirations to me.

I would like to thank Professor Paul Fieguth, Professor George Freeman, and Professor Murat Uysal for serving on my Ph.D. Committee, as well as for many helpful comments and insightful questions that helped to develop this work. I am grateful to Professor Xiaolin Wu for his commitment and service in the examining committee. I am also indebted to Professor Simon Yang, Professor Edward Vrscay, Professor Liang-liang Xie, and Dr. Da-ke He for their invaluable support to my fellowship applications.

It has been a great pleasure to be a part of the Multimedia Communications Laboratory. My sincere appreciation is extended to all friends and co-workers here for their helpful interaction and wonderful friendship over the years. Special thanks go to Dr. Haiquan Wang for his contributions to the work in Chapter 6. I am also grateful to Dr. Alexei Kaltchenko, Dr. Longji Wang, Dr. Wei Sun, and Dr. Xudong Ma for their expertise and friendship, and to James She for fruitful collaborations that we have had. I appreciate the enthusiastic helpfulness of the departmental support staff, particularly Wendy Boles, Lisa Hendel, Betty Slowinski, Paul Ludwig, Fernando Rivero Hernandez, and Philip Regier at all times.

Finally, I thank my wife, Suhuan Wang, for supporting me throughout the long path of my academic endeavor. This work could not have been accomplished without her love, support, and understanding.

To Suhuan

and

our lovely daughter Yunfeng and adorable son Danfeng

Contents

List of Tables

List of Figures

Abbreviations

AVC	Advanced Video Coding
CABAC	Context Adaptive Binary Arithmetic Coding
CAVLC	Context Adaptive Variable Length Coding
CBP	Coded Block Pattern
CD-ROM	Compact Disk-Read Only Memory
CIF	Common Intermediate Format
DCT	Discrete Cosine Transform
DV	Digital Video format, official name IEC 61834
DVD	Digital Video Disk
DWT	Discrete Wavelet Transform
HDQ	Hard Decision Quantization
HDTV	High Definition Television
HOS	Head of States
IEC	International Electrotechnical Commission
ISDN	Integrated Services Digital Network
ISO	International Organization for Standardization
ITU	International Telecommunication Union
ITU-T	ITU-Telecommunication Standardization Sector
JPEG	Joint Photographic Experts Group
MPEG	(ISO/IEC) Moving Picture Experts Group
NTSC	National Television System Committee TV System
PAL	Phase Alternating Line TV System
RD	Rate Distortion
SDQ	Soft Decision Quantization
UEG	Unary Exp-Golomb code
VCEG	(ITU-T) Video Coding Experts Group
VLC	Variable Length Coding
VQ	Vector Quantization

Chapter 1

Introduction

Broadly speaking, this thesis addresses some data compression problems in a practical multimedia system. As shown in Figure 1.1, the system involves a front device, an end device, and a connection in the between through channels or storage media. A conventional system setting for researching on video compression is the pair of encoder and decoder, assuming abundant computation power for encoding, limited computation power for decoding, and no diversity for spatial and temporal resolutions. Under this circumstance, a critical question is what the best RD trade-off is, which is the first problem to be tackled in this thesis. Furthermore, if we consider the spatial resolution diversity between the capturing unit and the displaying unit, there is a transcoding problem, which involves converting the spatial resolution for a compressed source. This transcoding task with spatial resolution conversion motivates the second major work in this thesis for image/video down-sampling in the DCT domain.

Lossy video compression under the conventional system setting with abundant encoding power generally adopts a hybrid structure shown in Figure **??**, where several different compression techniques such as motion prediction, transform, quantization, and entropy coding are employed together. In general, this is referred to as hybrid video compression [20, 55, 63]. This structure follows an intuitive understanding of video data about the temporal redundancy (similarity between

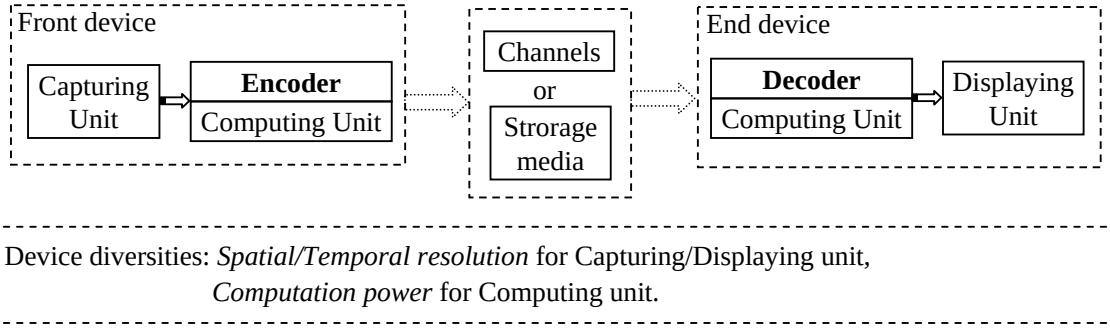


Figure 1.1: A multimedia system. Video compression in a practical multimedia system may be customized by different system settings such as its device diversities and the data delivery method. Conventional hybrid video compression assumes only the pair of encoder and decoder, overlooking the device diversities and the data delivery method. Transcoding considers the spatial resolution diversity, and/or the temporal resolution diversity, and/or channel bandwidth diversity through a network. Distributed video coding addresses the computation power diversity, technically speaking.

frames), the spatial redundancy (similarity between neighboring pixels), the psychovisual redundancy (limited sensitivity to spatial details by human eyes). Yet, it is still the most effective way for lossy video compression and has been adopted in all lossy video coding standards in the industry[63]. In this thesis, we will study the best rate distortion performance by hybrid video compression with compatibility to industrial standards.

Future research may also be well pictured in the multimedia system shown in Figure 1.1. The first is still spatial resolution conversion, but with a focus on handling motion re-prediction, which is not handled in this thesis. The second is temporal resolution conversion, which deals with the temporal resolution diversity. The third is to investigate how the computation power may be allocated between the front device and the end device in a flexible way, as to be discussed in details in the last chapter.

1.1 Thesis Motivations

Work in this thesis is mainly motivated by a desire to answer the following questions in the multimedia system shown in Figure 1.1.

1. What is the best RD coding performance for hybrid video compression?

   Ever since digital video was invented, video compression has been an essential part in any of its applications because of the enormous volume of video data[64, 13]. As digital video has become a ubiquitous and essential component of the entertainment, broadcasting, and communications industries, there is a never ending pursuit of more bandwidth/storage space for accommodating the explosively growing video data. This is fueling the demand for video compression to pursue the possibly best compression efficiency.

   Video compression generally assumes four types (temporal, spatial, psychovisual, and statistical) of redundancy, leading to a hybrid coding structure[40], as shown in Figure 1.2. The hybrid structure consists of four coding parts, i.e., motion compensation, transform, quantization, and entropy coding. Because the quantization part introduces permanent information loss to video data, hybrid video compression is usually categorized as lossy data compression. The theory that studies the theoretical limits for lossy data compression is called rate distortion theory [1]. Given an information source, the best coding efficiency that a compression method can achieve is characterized by the so-called rate distortion function, or equivalently distortion rate function [1]. Therefore, the fundamental trade-off in the design of a video compression system is its entire rate distortion performance.

2. What is the best RD coding performance an H.264-compatible codec can achieve?

   Video coding standards provide a solid base for the development of digital video industries by promoting worldwide interoperability. Therefore, our

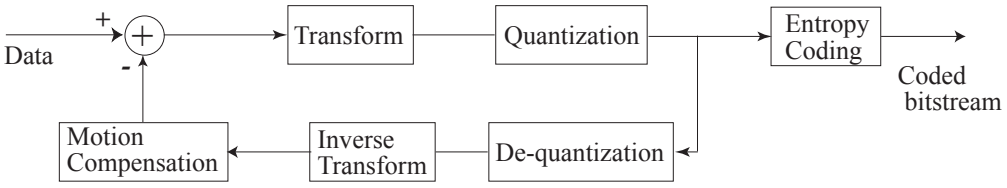


Figure 1.2: Illustration of a hybrid coding structure with motion prediction, transform, quantization and entropy coding.

study on the best RD coding performance will be within a standard coding scheme, i.e., to maintain compatibility with an industrial coding standard.

H.264, the newest hybrid video compression standard [79], has proved its superiority in coding efficiency over its precedents, e.g., it shows a more than 40% rate reduction over H.263 [77]. Meanwhile, from an engineering perspective, it is known that H.264 has utilized up-to-date technologies to improve its coding efficiency for each individual coding part from quarter-pixel motion prediction to complex binary arithmetic coding. It is interesting to see how much RD theoretic studies can help to further improve the coding performance for H.264 by jointly designing the whole hybrid coding structure.

3. How to construct efficient DCT-domain down-sampling methods for image/video transcoding?

   As wireless and wired network connectivity is rapidly expanding and the number of network users is steadily increasing, there has been a great momentum in the multimedia industry for supporting content display in diverse devices all over the network [52]. A big challenge, however, is the great diversity of devices with various display resolutions from full screen computers to small smart phones. This leads to researches on transcoding, which involves automatic reformatting. Particularly, image/video transcoding in the DCT domain with spatial resolution conversion has attracted increasing attentions[69], because most image/video data to be shared over the network are originally captured with a high resolution and coded using a transform technique of DCT, e.g., MPEG, JPEG, DV, etc.

1.2 Thesis Contributions

Contributions in this thesis are summarized as follow:

- A joint design framework for optimizing RD trade-off in hybrid video coding. Based on SDQ instead of conventional HDQ, we have proposed an RD optimization framework for jointly designing motion compensation, quantization, and entropy coding by minimizing the actual RD cost. The framework includes three algorithms, i.e., SDQ, residual coding optimization, and overall joint optimization, with them embedded in the indicated order. The framework may be applied to any hybrid video coding scheme by developing the three algorithms, particular the SDQ algorithm, according to given coding syntaxes.

- SDQ design based on CAVLC and SDQ design based on CABAC. In general, different entropy coding methods require different algorithms for SDQ. Depending on the entropy coding method involved, the problem of designing algorithms for optimal or near optimal SDQ in conjunction with that specific entropy coding method could be very challenging, especially when the involved entropy coding method is complicated. In some cases, for example, SDQ for GIF/PNG coding where the entropy coding methods are the Lempel-Ziv[84] [85] algorithms, the SDQ design problem is still open [35]. Fortunately, in this thesis, we are able to solve the design problems of SDQ in conjunction with CAVLC and CABAC, respectively. It is shown that given quantization step sizes, the proposed SDQ algorithms based on CAVLC and CABAC, respectively, achieve near-optimal residual quantization in the sense of minimizing the actual RD cost.

- A design framework for downsampling compressed images/video frames with an arbitrary ratio in the DCT domain. We first derive a set of DCT-domain down-sampling methods, which can be represented by LTDS in the DCT domain, and show that the set contains a wide range of methods with various

complexity and visual quality. Then, based on a pre-selected spatial-domain method, we formulate an optimization problem for finding an LTDS to approximate the given spatial domain method in the DCT domain for achieving the best trade-off between visual quality and complexity. By selecting a spatial-domain reference method with the popular Butterworth lowpass filtering and cubic B-spline interpolation, the proposed framework discovers LTDSs with better visual quality and lower computational complexity when compared with state-of-the-art methods in the literature. The obtained LTDSs will make a good application to transcoding non-predictively coded image/video such as JPEG or DV because of the good visual quality and high computational efficiency.

1.3 Thesis Organization

Chapter 2 presents a generic overview of hybrid video compression. In the first section, we review the four coding components in a typical hybrid coding structure, i.e., motion compensation, transform, quantization, and entropy coding. Since practices of data compression take root in Shannon's information theory[47], the discussion is intended to explain some underlying principles for those four coding parts from an information theoretic point of view. However, the theoretic discussion is limited to an introductory level. Some other discussions are presented from an algorithm design point of view, explaining corresponding state-of-the-art techniques and how they can be applied. Then, the next section introduces the development of video coding standards from the early MPEG-1 to the newest H.264 (also refereed to as MPEG-4, part-10) as background material and motivations to our study on RD optimization for video compression. Essentially, the development of those video coding standards shows that each individual coding part in the newest coding standard H.264 has been well designed to achieve superior coding performance using the state-of-the-art technologies. Optimization of an individual part alone will unlikely bring much improvement. This motivates our joint design framework

for hybrid video compression. Finally, the last section is devoted to details of the newest standard H.264, based on which we will develop algorithms for applying our proposed joint design framework to achieve the best coding performance while maintaining compatibility with H.264.

Chapter 3 presents the RD optimization framework for hybrid video compression. We begin with a brief survey on related work in the literature, highlighting the difficulty of using the actual RD cost in conventional RD optimization approaches. To tackle this issue, we discover an SDQ mechanism based on a universal fixed-slope lossy coding scheme. Using SDQ instead of the conventional HDQ, we then establish an RD optimization framework, which allows us to jointly design motion compensation, quantization, and entropy coding by minimizing the actual RD cost. Specifically, in the second section, we review the universal fixed-slope lossy coding scheme and apply it to optimizing hybrid video compression, obtaining SDQ. Based on the idea of SDQ, in the third section, we then formulate an RD optimization problem mathematically and derive an iterative solution, obtaining a generic RD optimization framework. In general, development of the residual coding optimization algorithm and the overall joint optimization algorithm is not directly related to specific coding syntaxes, except that they are based on the SDQ algorithm, which needs to be developed based on given coding syntaxes. Therefore, in the third section, we describe algorithms for residual coding optimization and overall joint optimization in details, leaving the SDQ design to be tackled when concrete coding syntaxes are given in a real application of the proposed RD framework.

In Chapter 4, we discuss the application of the joint design framework to optimizing RD trade-off for H.264 baseline profile encoding. As the residual optimization algorithm and the overall joint optimization algorithm have been described in Chapter 3 in details, we are focused on the SDQ design based on the entropy coding method in H.264 baseline profile, i.e., CAVLC. Specifically, a detailed review of CAVLC is presented in the first section. Then, in the second section, we exam the computation of the RD cost, based on which we construct a graph structure so

that the RD cost can be computed in an additive manner. As a result, the additive computation of the RD cost enables us to use dynamic programming techniques to search for quantization outputs to minimize the actual RD cost, yielding an SDQ design based on CAVLC. The SDQ design is then embedded into the residual coding algorithm, which is further called in the overall joint optimization algorithm. In the third section, we provide experimental results for implementing these algorithms based on H.264 reference codec Jm82.

Sharing a similar organization as that in Chapter 4, Chapter 5 is focused on the application of the joint design framework to optimizing H.264 main profile encoding with CABAC. The chapter begins by reviewing CABAC. Similarly, in the second section, we exam the RD cost computation based on CABAC. Compared with SDQ based on CAVLC, SDQ based on CABAC is more complicated because of the adaptive context updating in CABAC. To tackle this problem, we decompose the SDQ algorithm into two steps: the first step is SDQ with fixed probability context and the second step is context updating with fixed quantization outputs. The second step is straightforward. For the first step, a graph is constructed based on context modeling in CABAC. Finally, we implement the resulting algorithms and report experimental results in the third section.

Chapter 6 presents the designing framework for down-sampling images in the DCT domain. In the first section, extensive discussions are presented on pixel-domain down-sampling, which is the basis for designing DCT-domain down-sampling algorithms. Then, in the second section, we review other DCT-domain down-sampling methods in the literature. Based on these discussions, the third section derives a linear transform with double-sided matrix multiplication for down-sampling in the DCT domain. The linear transform is equivalent to a concatenation of inverse DCT, pixel-domain down-sampling, and DCT. The next section discusses the visual quality measurement for down-sampled images. Then, we establish a complexity model for LTDS, based on which a joint optimization problem is formulated mathematically to find the optimal LTDS for DCT-domain down-sampling by optimizing

trade-off between visual quality and computational complexity. The problem is solved using a multiple-layer neural network structure and a structural learning with forgetting algorithm. We conduct some experiments and provide comparative results in the last section.

Finally, Chapter 7 concludes the thesis and discusses future research.

Chapter 2

Hybrid Video Compression Overview

An important feature for lossy video compression, as discussed in the introduction chapter, is a hybrid coding structure [78]. In fact, all lossy video coding standards, from the earliest MPEG-1 to the newest H.264[72], employ this hybrid coding structure. In this chapter, we first review the basic structure of hybrid video coding. Then, we briefly introduce the development of those lossy video coding standards for their main technical features, showing how coding techniques for individual parts in the hybrid structure evolve through the development. Finally, we present a detailed review of H.264, since one of the main objectives in this thesis is to maintain compatibility with H.264 while optimizing its RD trade-off.

2.1 Hybrid Coding Structure

As shown in Figure 1.2, there are four coding parts in the hybrid coding structure, i.e., motion compensation, transform, quantization, and entropy coding. This section reviews individual coding parts in the hybrid coding structure for their underlying principles and some design issues.

2.1.1 Motion Compensation

Video signals display a distinct kind of redundancy called temporal redundancy, i.e., the high similarity between neighboring frames. While image data are well known for spatial redundancy among neighboring pixels, for video compression higher similarities are often observed among nearby frames than within a single frame. In fact, the major difference between still image compression and video coding is the temporal redundancy processing in the latter [40].

Motion compensation reduces temporal redundancy by estimating the current frame from reproductions of previously coded frames. A typical scenario is that an object moves from one location to another location. Once the object is encoded in one frame, its appearance in all the consecutive frames can be well represented with two factors, i.e., its shape and the displacement. Motion compensation that allows arbitrary shapes is conceptually advanced since an object may be of any shape[65]. However, it turns out that the coding performance of object-based motion compensation is much worse than that of a block-based coding scheme [63], because of the high rate for coding the shape. Thus, block-based motion compensation is more widely used in video compression standards. An important factor for block-based coding is the block size. In general, a small block size will lead to more motion vectors, which means more overhead bits. However, it also means a better prediction. H.264 uses square/rectangle blocks for motion compensation with various block sizes, e.g., 16×16, 16×8, 8×16, 8×8, 8×4, 4×8, 4×4, resulting in more flexibility for this new standard to achieve superior coding efficiency.

Another important factor for motion compensation is the prediction accuracy. In the early standard H.261, motion compensation is conducted on the original pixel map, so-called full-pixel prediction. The newest H.264 supports up to $\frac{1}{4}$-pixel accuracy for the luma component. Because samples at sub-pixel positions do not exist, they are created by interpolation in the reference frames. In general, the higher the prediction resolution is, the more effective motion compensation will be. However, studies by Girod in [21] show that the gain by using higher

prediction accuracy is limited in the sense that it becomes very small beyond a critical accuracy. It is suggested that $\frac{1}{2}$-pixel accuracy be sufficient for motion compensation based on videophone signals, while $\frac{1}{4}$-pixel accuracy be desirable for broadcast TV applications [21].

Multihypotheses prediction is another hot topic for motion compensation, where a prediction is obtained by a linear combination of more than one hypothesis[1] from multiple reference frames. Flierl et al. derived a performance bound [17] for averaging multiple hypothesis by extending the wide-sense stationary theory of motion-compensated prediction in [20]. It shows that the gain is limited even when the number of hypotheses goes to infinite. It is suggested that two hypotheses provide the most efficient solution, leading to applications of the so-called B-frame design. (See [16] for B-frame design in H.264.) In addition, Girod [22] points out that the introduction of bi-direction is more efficient than doubling the prediction accuracy.

With all these flexibilities for motion compensation, i.e., the block size, accuracy, multiple references, it is of natural interest to think of a criterion for finding an optimal setting of these parameters. Early in 1987, Girod [20] proposed a rate-distortion analysis of motion compensation by relating the power spectral density of the prediction error to the prediction accuracy. Later, the analysis was extended to multihypotheses motion compensation [22, 17], providing guidance for using P-frames (one reference) and B-frames (two references). Corbera and Neuhoff [8][7] also developed a theoretical framework for selecting the best prediction accuracy in the sense of minimizing a joint cost of the prediction error and the coding rate of motion vectors. The above theoretical analyses provided valuable insights in the underlying mechanism of motion compensation.

Beside the above analytical studies, many experimental methods have been developed and used, e.g., motion compensation in H.264 is optimized as follows

[1]The term of hypothesis used here means one estimation of a given pixel block based on a given reference frame.

[77],

$$\arg\min_{\mathbf{v}} J_p = ||\mathbf{x} - \mathbf{p}(\mathbf{v})|| + \lambda \cdot R(\mathbf{x}), \tag{2.1}$$

where $\mathbf{x}$ is the pixel data, $\mathbf{p}$ is the prediction, $\mathbf{x}$ is the corresponding motion vector, and R is the rate for coding the motion vector. This optimization works fairly well. However, the compression distortion actually comes from quantizing the residual $\mathbf{x} - \mathbf{p}(\mathbf{v})$, instead of the residual itself. Meanwhile, the residual coding rate is not considered. Therefore, the cost function in (3.1) is not based on the actual distortion and the entire rate. In other words, (3.1) does not minimize the actual rate distortion cost, suggesting that there is still space for further optimization.

There are still many problems open for motion compensation. E.g., motion compensated prediction assumes translational motions, i.e., the current picture can be locally modeled as a translation of some previously coded pictures. In case of rotations or shape changing, this is a fundamental defect. The zooming operation of the camera is also a problem. Studies in [20, 22] suggest that the performance of motion compensation is essentially limited by the assumption of translational motions. Another problem for conventional motion compensated prediction is its high complexity, which results in a slow encoder. This actually motivates the research on distributed video coding, which has attracted a lot attention in the video coding community recently (see [23] and reference therein).

2.1.2 Transform

Transform coding works by converting data into a transform domain. Ever since its invention in early 1960's, transform coding has been widely used for lossy compression of video, image, and audio[25] [41]. The motivation for using transform coding is to decorrelate signals so that the outputs can be efficiently coded using simple techniques such as scalar quantization. In general, transform coding is always linked to scalar quantization [19, 26].

Among many block-based transforms, the most popular one is the discrete cosine transform (DCT), which has been adopted in all lossy video coding standards.

While 8×8 DCT is used in early standards, the H.264 uses a 4×4 DCT, which gives better coding efficiency and less block effect. As suggested from [49], the coding gain for using a small block size comes from the reduced interblock correlation.

Another popular transform is the discrete wavelet transform(DWT), which is based on the whole image. Although DWT has proved a big success in still image compression, it tends to be less attractive than the DCT in the case of video compression. The image-based DWT transform leads to design problems for block-based motion compensation. E.g., the mode selection for motion compensation requires to compute the rate cost for coding residuals. For any given block, there are a few modes. It is impossible to find the optimal mode for all blocks using the image-based DWT. However, DWT still makes its way into the MPEG-4 Visual standard, as an alternative option besides DCT.

From the correlation point of view, however, the concatenation of motion compensation and transform coding is non-optimal. Intuitively speaking, the more effective is motion compensation, the less correlated are the residuals, thus the less interesting for transforming the residual to the frequency domain. Studies in [20, 22, 46] pointed out that residuals after motion compensation are only weakly correlated.

From the information theoretic point of view, the transform plus scalar quantization and entropy coding method is questionable too. The DCT transform tends to generate coefficients with Gaussian distributions when the block size is large, which may be justified by applying the central limit theorem. Particularly, Eude et al. showed by mathematical analyses that DCT coefficients of images could be well modeled with a finite mixture of Gaussian distributions [14]. Information theory shows that the rate distortion function of a stationary source achieves its upper bound with Gaussian distributions [1], indicating that Gaussian source is the most difficult for lossy compression either by vector quantization or by a scheme with scalar quantization and entropy coding [26]. The fact that a small block size gives a better performance for using DCT transform possibly indicates that DCT

transform in the hybrid structure is of much interest for reconsideration.

2.1.3 Quantization

The application of quantization to video compression is inspired by some cognitive studies on human visual systems. Human visual systems show excellent robustness in extracting information from video signals [59]. Bioelectrically, the human eye's response to spatial details is limited. Thus, a certain amount of distortion may be introduced into video signals while a human observer would not notice it. Furthermore, the human visual system allows a wide range of even noticeable distortion while it is still able to obtain critical information from the video signals. In other words, there exist much *psychovisual redundancy* in image/video signals. From information theoretical point of view, this psychovisual redundancy makes it possible to balance bandwidth and distortion according to given channel conditions, leading to the application of quantization.

Most video compression designs use scalar quantization, which is basically a simple arithmetic operation to shrink the dynamic range of inputs [63] [55]. It is a hard decision based operation in the sense that the quantization output for a given input is directly computed from the input itself and a quantization step size. Its major merit is the simplicity, which is well demonstrated by the complexity of vector quantization (VQ) as the rival. Information theoretic analysis on source coding with a fidelity criterion shows that there exists an unbeatable bound, which is characterized by the distortion-rate function of a source with respect to a distortion measure [1, 83]. This Shannon lower bound is approximately achievable by VQ. Gray proved that VQ can perform arbitrarily close to the Shannon lower bound [26]. Unfortunately, designing such optimum quantizers can be very difficult [18]. Meanwhile, the high complexity due to the exponentially increasing vector space makes it almost infeasible for many real-world applications [19].

On the other hand, comparative studies between optimized VQ and scalar quantization discover interesting results, supporting scalar quantization [19, 83, 26].

Gray showed that the optimum vector quantizer should have a uniform density of reproduction levels in the vector space when some extra effort is allowed to code quantization outputs with lossless codes [26]. Under such a circumstance, one is using entropy to measure rate. The quantizer is designed to minimize the entropy of its outputs, while a lossless algorithm is used to achieve a coding rate as close to the entropy as it can. In general, this result suggests a promising combination of scalar quantization and entropy coding.

Theoretical studies based on the rate distortion theory provide further support for using scalar quantization [36, 24]. Consider a generic scheme of quantization plus entropy coding as discussed above. The performance of a quantizer Q can be measured by two quantities, i.e., quantization distortion and the entropy of quantization outputs H_Q. An essential result obtained by Gish and Pierce in [24] showed that for small distortions (high rate) and a memoryless source with a smooth marginal density, H_Q of a scalar quantizer exceeds the Shannon lower bound $R_{SLB}(D)$ by around 0.255 bit/sample, i.e.,

$$H_Q - R_{SLB}(D) \approx 0.255.$$

Denote the rate-distortion function for the source as $R(D)$. Considering that the Shannon lower bound is strictly less than $R(D)$ and that the above approximate equality is very close to equality, we can have [26]

$$H_Q \leq R(D) + 0.255,$$

which is very encouraging for combining scalar quantizer and lossless codes. It indicates that the combination can perform almost within one quarter of a bit of the Shannon optimum.

The above analysis well explains the wide application of scalar quantization. However, it does not mean that it is of no interest to look for optimum quantizers, particularly in the low bit rate case. In fact, it has been shown that scalar quantization is not good in the case of low bit rate[36, 26, 66], e.g., 0.5 bit/sample. The gap between H_Q and the Shannon bound, which may be ignorable for high bit

rate applications, becomes a bottle-neck for using scalar quantizers in low bit rate systems.

There are two methods to deal with this gap. One is to use VQ [68, 18]. For a k-dimensional vector quantizer, the gap between the kth order entropy of the quantized signal and the Shannon lower bound is approximated as [36, 26],

$$\lim_{D\to 0}[H_Q^{(k)} - R_{slb}^{(k)}(D)] \geq \frac{1}{2}\log\frac{e(\Gamma(1+k/b))^{b/k}}{1+k/b},$$

where b is the distance norm for measuring distortion, e.g., $b = 2$ for using Euclidean distance. Calculation of the right side with $b = 2$ shows the gap as 0.255, 0.221, and 0.178 for $k = 1$, $k = 2$, and $k = 4$, respectively. Obviously, a rate gain is achievable by coding the k-dimensional quantization outputs. However, there is a big complexity issue because the size of the vector space grows exponentially with the dimension.

Another method is to introduce soft decision quantization [32], by which we mean that quantization outputs are generated based on a rate distortion cost for an array of inputs, as to be discussed later. An intuitive interpretation of soft decision quantization is to adapt quantization outputs to the coding context of a given lossless coding algorithm. For hard decision quantization, the output is totally unrelated to the entropy coding part. Under such a circumstance, the best rate performance of the whole scheme is bounded by the entropy of quantization outputs. Then the gap between the entropy and the Shannon lower bound is an inevitable loss. However, studies in [36] show that the original entropy bound can be exceeded by optimizing quantization outputs with respect to the following lossless coding. As a result, the coding rate of the lossless algorithm will asymptotically approach the optimum given by the rate-distortion function.

2.1.4 Entropy Coding

Intuitively, designs of prediction, transform, and quantization are based on cognitive modeling of video signals, i.e., they may be regarded as aiming at temporal, spatial,

and psychovisual redundancy in video data, respectively. Entropy coding, on the other hand, is independent of specific data characteristics and is developed based on mathematical data modeling, specifically, Shannon's information theory [9]. In Shannon's information theory, entropy means the amount of information presented in a source, which is quantitatively defined as the minimum average message length that must be sent to communicate the value of the source to a receiver.

While multimedia compression is usually lossy, the entropy coding part is lossless. In a hybrid video coding structure, the information loss comes from the quantization part only. After quantization, entropy coding is designed to precisely represent quantization outputs and other overhead symbols with possibly minimum number of bits. According to Shannon's source coding theorem, the optimal number of bits for coding a source symbol is $-\log_2 p$, where p is the probability of the input symbol[2]. An entropy coder seeks for the minimal number of bits for coding a given set of symbols [26].

The two most popular entropy coding methods are Huffman coding [28] and arithmetic coding [51]. The basic idea of Huffman coding is to encode a symbol that has higher probability with a less number of bits, which exactly follows Shannon's guideline of $-\log_2 p$. The problem, however, is that $-\log_2 p$ may not be an integer, leading to an loss of coding efficiency by up to 1bit/symbol. For example, a source symbol with $p = 0.248$ would transmit 2.01 bits of information, but it consumes 3bits if Huffman coding is used. This efficiency loss comes from the fact that Huffman coding can only assign an integer number of bits to code a source symbol.

Arithmetic coding is generally superior to variable length coding such as the Huffman coding because it can adapt to symbol statistics and assign a non-integer number of bits to code a symbol. The main idea of arithmetic coding is to treat a sequence of symbols as one input and to generate one unique codeword accordingly. So, there is no explicit code-book. Instead, the whole sequence of symbols is mapped

[2]In general the base should be the number of symbols used for generating output codes. But the base 2 is always used in this thesis so that the coding length is measured by bits.

to a point on the real axis, whose binary representation is then taken as the coding output, and the length of the binary representation is $-\log_2 p$ with p being the probability of the whole sequence. E.g., for an I.I.D source with alphabet set $\{x_1, x_2, x_3\}$ and probability model $p(x_1) = \frac{1}{2}$, $p(x_2) = \frac{1}{3}$, $p(x_3) = \frac{1}{6}$, s sequence of $(x_1x_3x_2x_2x_3x_3x_1x_2)$ would result in a codeword of length ceil$(-\log_2(\frac{1}{2}\frac{1}{6}\frac{1}{3}\frac{1}{3}\frac{1}{6}\frac{1}{6}\frac{1}{2}\frac{1}{3}))$ by arithmetic coding. This corresponds to a coding rate of 1.875bit/symbol, while the rate for Huffman coding is 2.125bit/symbol, which is computed as $(1+3+2+2+3+3+1+2)/8$ because x_1, x_2, x_3 accord to codewords of 1bit, 2bits, and 3bits, respectively. Certainly, there is a price for arithmetic coding to pay for its high compression efficiency, i.e., the high computational complexity.

Since an entropy codec is designed based on a mathematical model, the coding efficiency of an entropy codec in a real-world application is largely dependent on how well we can establish a mathematical model for the data to be compressed. Shannon's source coding theorem establishes a relationship between the symbol probability and the corresponding coding bits. In order to find the optimal representation, i.e., the minimal number of bits, the probability distributions of all symbols are required to be known, which unfortunately is not true for most real world applications. The solution is to estimate the distributions. In general, this is a big challenge for designing entropy coding methods. It requires complicated design and extensive computation. E.g., extensive experiments are conducted to study the empirical distributions of various syntax elements in H.264. Eventually, there are over 400 context models developed and complicated criteria are defined for context selection in the CABAC method[51].

2.2 Hybrid Video Coding Standards

International standards for video compression have played an important role in the development of the digital video industry. Since early 1980's, many standards have been developed. Each standard is the result of many years of work by a lot of

people with expertise in video compression. It is interesting to have a look at the development of these standards.

2.2.1 MPEG-1

The first video coding standard that proved a great success in market was MPEG-1 [13], developed by ISO/IEC MPEG. MPEG-1 was designed for bit rate up to 1.5 Mbps. This was based on CD-ROM video applications. Today, it is still a popular standard for video on the Internet. It is also the compression standard for VideoCD, the most popular video distribution format throughout much of Asia.

Technically, MPEG-1 is very simple if compared with today's standard. However, it has utilized all the main coding techniques for hybrid video coding such as bi-directional inter-frame coding, DCT, variable length coding, etc., except that these techniques in MPEG-1 have not been developed as well as they are in latter standards. For motion prediction, MPEG-1 supports the three main types of frames, i.e., I-frame for intra prediction, P-frame for inter prediction, and B-frame for bi-directional prediction. The block partition in I-frames is 8×8, while the block size for inter prediction in P-frames is fixed as 16×16. Also, the prediction in MPEG-1 is based on full-pixels, while later on it advances to support half-pixel in MPEG-2, and quarter-pixel in H.264. DCT in MPEG-1 uses an 8×8 block size. For quantization in MPEG-1, there is one step size for the DC coefficient, and 31 step sizes for the AC coefficients. The 31 step sizes take the even values from 2 to 62. For AC coefficients of inter-coded blocks, there is also a dead-zone around zero. Finally, entropy coding in MPEG-1 uses a simple scheme of concatenating run-length coding with variable length coding (VLC). A small VLC table is defined for most frequent run-level pairs, while other run-level combinations are coded as a sequence of 6-bit escape, 6-bit codeword for run, and 8-bit codeword for levels within $[-127, 127]$ or 16-bit codewords for other levels.

2.2.2 MPEG-2

MPEG-2 [64] was developed soon after MPEG-1 because it turned out that MPEG-1 could not provide a satisfactory quality for television applications. MPEG-2 was then designed to support digital television set top boxes and DVD applications.

Ever since it was finalized in November 1994, MPEG-2 has become a fundamental international standard for delivering digital video. The worldwide acceptance of MPEG2 opens a clear path to worldwide interoperability. Today, MPEG-2 plays an important role in the market and it will continue to do the same in the near future according to some market forecast such as the Insight Research Cooperation for MPEG-2 related products. All of the industries who target digital video services have to invest in MPEG-2 applications. MPEG2 based video products are developed for a wide range of applications, as to name a few in the following.

1. DVD: As a new generation of optical disc storage technology, DVD offers an up to 10G storage space for MPEG-2 video distribution. Ever since its introduction, DVD has become the most popular MPEG-2 based video product.

2. HDTV: MPEG-2 compression is used in HDTV applications to transmit moving pictures with resolution up to 1080×1920 at rate up to 30frame/second (requiring 20MHz bandwidth) through 8MHz channels.

3. Digital Camcorders: Originally, all digital camcorders use the Digital Video (DV) standard and record onto digital tape cassettes. However, the latest generation of camcorders turns to use MPEG2 because it provides a high compression with high quality. Video data can be recorded directly onto flash memory or even to a hard disk. While transferring video files from a tape is slow because it requires real time play-back, a flash card/DVD/hard disk provides a much faster access to the video data.

Figure 2.2 illustrates the motion compensation procedure in MPEG-2. Basically, motion compensation design in MPEG-2 is still at a primitive stage. Many issues

such as the block size and prediction accuracy were not effectively addressed. In particular, motion compensation in MPEG-2 is based on a fixed size of 16×16, which leads to poor prediction when there are a lot of details in images. The prediction accuracy is fixed at half-pixel, while studies by Girod [22] show that quarter-pixel accuracy is required for efficient motion compensation when distortion is small.

MPEG-2 utilizes 8×8 DCT. The DCT design is as follows [63],

$$\mathbf{Y} = \mathbf{A} \cdot \mathbf{X} \cdot \mathbf{A}^{\mathrm{T}} \tag{2.2}$$

$$\mathbf{X} = \mathbf{A}^{\mathrm{T}} \cdot \mathbf{Y} \cdot \mathbf{A} \tag{2.3}$$

where $\mathbf{A}$ is an $N \times N$ transform matrix with its element

$$A_{i,j} = C_i \cos \frac{(2j+1)i\pi}{2N} \text{ where } C_0 = \sqrt{\frac{1}{N}}, \quad C_i = \sqrt{\frac{2}{N}} (i > 0).$$

MPEG-2 uses $N = 8$. As shown in Figure 2.2, the 8×8 block is the fundamental unit for residual coding in MPEG-2.

Scalar quantization is applied to each 8×8 block of DCT coefficients in MPEG-2 with lower frequency coefficients taking smaller quantization step sizes and higher frequency components taking larger quantization step sizes. Specifically, an 8×8 weighting matrix is defined for inter blocks as follows,

$$\mathbf{w} = \begin{bmatrix} 16 & 17 & 18 & 19 & 21 & 23 & 25 & 27 \\ 17 & 18 & 18 & 21 & 23 & 25 & 27 & 29 \\ 18 & 19 & 20 & 22 & 24 & 26 & 28 & 31 \\ 19 & 20 & 22 & 24 & 26 & 28 & 30 & 33 \\ 20 & 22 & 24 & 26 & 28 & 30 & 32 & 35 \\ 21 & 23 & 25 & 27 & 29 & 32 & 35 & 38 \\ 23 & 25 & 27 & 29 & 31 & 34 & 38 & 42 \\ 25 & 27 & 29 & 31 & 34 & 38 & 42 & 47 \end{bmatrix},$$

and a quantization scalar q_s is defined as an integer within $[1, 31]$. Then the quantization syntax is specified in MPEG-2 by the following de-quantization equation,

$$\hat{c}_{ij} = u_{ij} \cdot q_s \cdot w_{ij}/8,$$

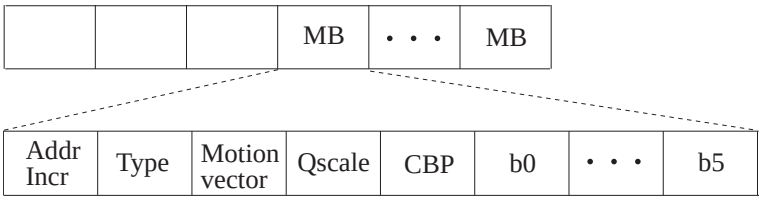


Figure 2.1: The syntax of macroblocks in MPEG-2.

where u_{ij} stands for a quantized coefficient, w_{ij} is the (i, j)th element in $\mathbf{w}$, and $\hat{c}_{ij}$ is the corresponding reconstruction. Quantization for intra blocks is slightly different. For an intra block, its DC components are quantized using one of 4 quantization step sizes, i.e., 1, 2, 4, 8, Accordingly, the 11-bit dynamic range of the DC coefficient is rendered to accuracy of 11, 10, 9, or 8 bits, respectively.

Each quantized coefficient in MPEG-2 is encoded as two parts, i.e., its absolute value and the sign. A set of variable length coding tables is designed to code the absolute values of quantized coefficients and other syntax elements. These tables are often referred to as modified Huffman tables, in the sense that they are not optimized for a limited range of bit rates. Coefficient signs are coded using fixed length codes with an underlying assumption that positive and negative coefficients are equally probable.

In summary, Figure 2.2 illustrates the hybrid coding procedure of MPEG-2. For a given macroblock, a motion vector is found by matching its 16×16 luma block with blocks in previously coded images, called reference frames. Predictions for both the luma block and two chroma blocks are computed based on this vector. Then, residuals are partitioned into 8×8 blocks and transformed using DCT. Scalar quantization is applied to the transform coefficients. Finally, variable length codes are used to encode the quantized coefficients.

2.2.3 MPEG-4/H.264

MPEG-2 was so successful that the MPEG working group aborted its work on updating MPEG-2 to MPEG-3. MPEG started to work on a new standard MPEG-4

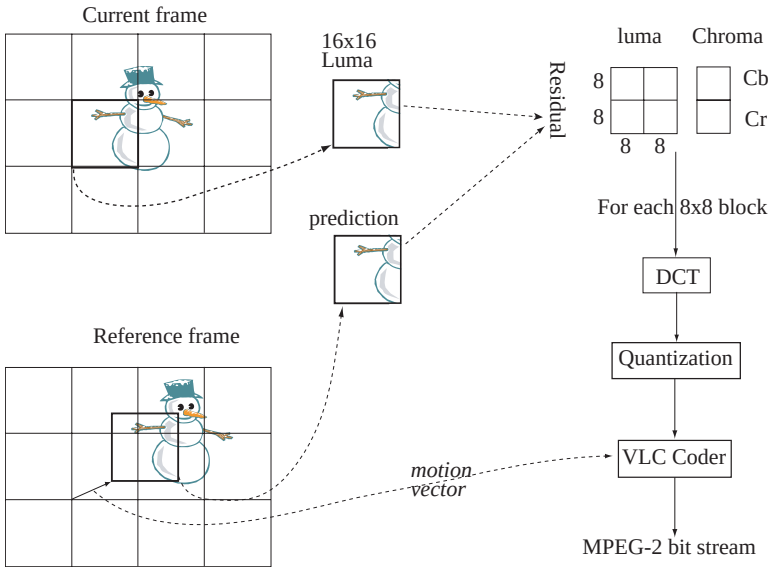


Figure 2.2: The hybrid encoding process of MPEG-2. Motion vector search is based on the 16×16 luma block with half-pixel accuracy. Residuals are divided into 8×8 blocks. They are transformed with 8×8 DCT and quantized. Finally, variable length codes are used to code the quantized coefficients.

[63] in 1993. Besides ISO/IEC MPEG, ITU-T VCEG is another working group who takes a leading place in video coding standard development. Its work is essentially focused on efficient video communications over telecommunication networks and computer networks. H.261 was the first successful standard for video-conferencing developed by VCEG [13]. It was designed for two-way video communications over ISDN, targeting data rates at multiples of 64kbps. It only supported two image resolutions, i.e., common intermediate format (CIF) and quarter CIF. In order to achieve lower bit rate compression, VCEG developed H.263, targeting data rate below 30kbps [64]. Many new technologies were then introduced into the standard, e.g., arithmetic coding. In general, these technologies required much more computation. However, advances in silicon technologies well compensated these requirements. The resulting coding performance won a big success for H.263. Finally, it was adopted by MPEG as the compression core of MPEG-4 Visual, also called MPEG-4, part-2. MPEG-4 Visual is designed to support both efficient video

compression and various content-based functionalities.

Besides some short-term effort on improving H.263, e.g., H.263+ and H.263++, VCEG started to develop an entirely new standard for low bit rate video compression in 1995. The outcome was the H.26L draft. The new standard had a narrower scope than MPEG-4 Visual. Essentially, it was focused on efficient coding for rectangular moving pictures. It showed a significant improvement on the coding performance over previous standards. In 2001, MPEG decided to adopt H.26L as the core of its advanced video coding design. Then, a joint video team was formed by experts from both MPEG and VCEG. The new standard, by the name ITU-T H.264 and ISO/IEC MPEG-4 Advanced Video Coding, was published in 2003. For simplicity, it is often referred to as H.264.

2.3 Detailed Review of the Newest Standard H.264

H.264 was finalized in 2003. As the most efficient video coding standard at this point, H.264 utilizes many advanced coding technologies, e.g., adaptive block size, quarter-pixel prediction accuracy, 4x4 DCT, arithmetic coding, etc.

2.3.1 The Great Potential of H.264

H.264 offers significantly higher coding efficiency than its predecessors. In general, it is reported to give twice the compression of MPEG-4 Visual, or triple the compression of MPEG2 [78]. The digital video industry shows a warm welcome to the new standard. E.g., an announcement from the Apple company said that the DVD forum has ratified H.264 to be included in the next generation High Definition DVD format. As shown in table 2.1, companies are active in developing products for H.264.

Table 2.1: H.264 codecs and vendors

Vendor	Encoder	Homepage
Apple	QuickTime for Tiger - N.A.	www.apple.com/macosx/tiger/h264.html
Harmonic Inc.	DiviCom Encoder (HW)	www.harmonicinc.com
Videosoft	H.264 encoder	www.videosoftinc.com/
HHI	JM 8.5 reference software	bs.hhi.de/ suehring/tml/
Dicas	mpegable AVC (free)	www.mpegable.com/show/mpegableavc.html
Mainconcept	Preview Encoder	www.mainconcept.com/h264_encoder.shtml
Modulus Video	SDTV, HDTV Encoder	www.modulusvideo.com/
PixelTools	Expert H.264	www.pixeltools.com/experth264.html
UBVideo	UBLive-264-C64 (Videoconferencing)	www.ubvideo.com/mainmenu.html
Media Excel	Softstream	www.mediaexcel.com/products.htm
LSILogic	H.264 VLE4000 (HW)	www.lsilogic.com/products/video_broadcasting/vle4000.html
Envivio	4Caster (HW)	www.envivio.com/products/4caster.html
Envivio	4Coder (SW)	envivio.com/products/4coder_se.html

Table 2.2: Compression performance for various video coding standards.

	Feature	Compression Performance			
MPEG-1	Poor quality				
MPEG-2	In the market	Resolution	Raw rate		Real rate
		352 × 288	34.8Mb/s	Set-top boxes	4Mb/s
		720 × 480	118Mb/s	DVD	9.8Mb/s
				SD-DVB	15 Mb/s
		1920×1080	712Mb/s	HDTV	80 Mb/s
H.263	Low rate applications	1.5 time compression of MPEG-2 [77]			
H.264	All applications	triple the compression of MPEG-2 [77]			

2.3.2 Hybrid Coding in H.264

Development of H.264 had a very clear target at its beginning, i.e., to utilize the great advances in silicon technologies to achieve possibly the best coding efficiency. H.264 is based on the same hybrid coding structure as MPEG-2. However, there are many significant improvements in detailed designs.

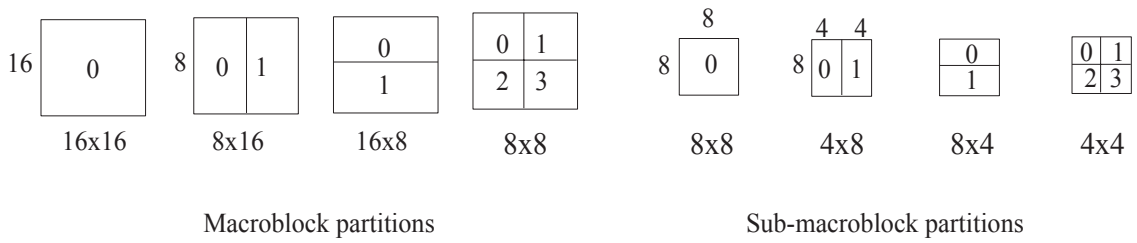


Figure 2.3: Block partitions in H.264.

Motion Compensation in H.264

While prediction in MPEG-2 is at the primitive stage, the prediction design in H.264 has been significantly improved. It allows various block sizes from 16×16 to 4×4 as shown in Figure 2.3. While a large block size is desirable for homogeneous regions, a small size makes it possible to catch details more efficiently.

The prediction accuracy for H.264 is $\frac{1}{4}$-pixel, which is suggested as the highest precision that is required in order to achieve optimal coding performance [22]. To compute the half-pixel samples, a 6-tap finite impulse response filter is designed with weights (1/32, -5/32, 20/31, -5/32, 1/32). Given samples in Figure 2.4, the half-pixel sample b is

$$b = (E - 5F + 20G + 20H - 5I + J)/32.$$

Then, quarter-pixel samples a, d, e are obtained by linear averaging as follows,

$$a = (G + b)/2, \qquad d = (G + h)/2, \qquad e = (b + h)/2.$$

Transform in H.264

H.264 uses the well-known discrete cosine transform (DCT) with a block size of 4×4 in its baseline profile and main profile. Specifically, the transform matrix is

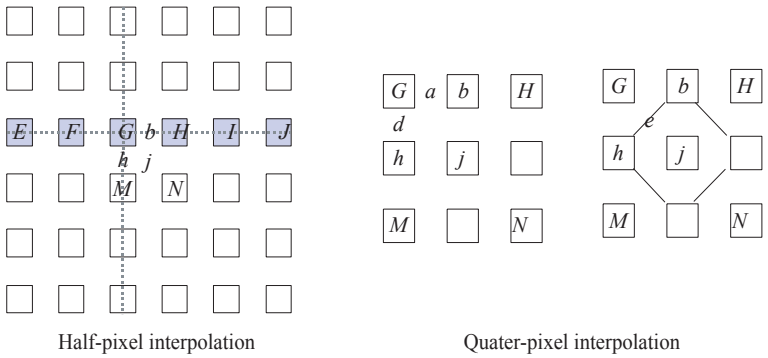


Figure 2.4: Interpolation for $\frac{1}{4}$-pixel motion compensation in H.264.

$$\hat{\boldsymbol{w}} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{\sqrt{2.5}} & \frac{0.5}{\sqrt{2.5}} & -\frac{0.5}{\sqrt{2.5}} & -\frac{1}{\sqrt{2.5}} \\ \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \\ \frac{0.5}{\sqrt{2.5}} & -\frac{1}{\sqrt{2.5}} & \frac{1}{\sqrt{2.5}} & -\frac{0.5}{\sqrt{2.5}} \end{pmatrix}.$$

To facilitate fast implementation with integer operations, a simplified transform matrix is obtained as

$$\boldsymbol{w} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1/2 & -1/2 & -1 \\ 1 & -1 & -1 & 1 \\ 1/2 & -1 & 1 & -1/2 \end{pmatrix},$$

by extracting a factor matrix $\boldsymbol{f}$ from $\hat{\boldsymbol{w}}$ as

$$\boldsymbol{f} = \begin{pmatrix} \frac{1}{4} & \sqrt{\frac{1}{10}} & \frac{1}{4} & \sqrt{\frac{1}{10}} \\ \sqrt{\frac{1}{10}} & \frac{2}{5} & \sqrt{\frac{1}{10}} & \frac{2}{5} \\ \frac{1}{4} & \sqrt{\frac{1}{10}} & \frac{1}{4} & \sqrt{\frac{1}{10}} \\ \sqrt{\frac{1}{10}} & \frac{2}{5} & \sqrt{\frac{1}{10}} & \frac{2}{5} \end{pmatrix},$$

with $\hat{\boldsymbol{w}}\,\boldsymbol{Y}\,\hat{\boldsymbol{w}}^{\mathrm{T}} = (\boldsymbol{w}\,\boldsymbol{Y}\,\boldsymbol{w}^{\mathrm{T}}) \otimes \boldsymbol{f}$ for any 4×4 matrix $\boldsymbol{Y}$ where the symbol $\otimes$ denotes element-wise multiplication between matrixes.

Quantization in H.264

Quantization in H.264 is achieved simply by a scalar quantizer. It is defined by 52 step sizes based on an index parameter $p = 0, 1, \cdots, 51$. The quantization step size for a given p is specified as

$$q[p] = h[p_{\text{rem}}] \cdot 2^{p_{\text{quo}}}, \tag{2.4}$$

where $p_{\text{rem}} = p\%6$ and $p_{\text{quo}} = \text{floor}(p/6)$ are the remainder and quotient of p divided by 6, and $h[i] \in \{\frac{10}{16}, \frac{11}{16}, \frac{13}{16}, \frac{14}{16}, \frac{16}{16}, \frac{18}{16}\}$, $6 > i \geq 0$.

For the purpose of fast implementation, quantization and transform in H.264 are combined together. Specifically, the factor matrix $\boldsymbol{f}$ is combined with the quantization step size. Suppose that the decoder receives the quantized transform coefficients $\boldsymbol{u}$ and the quantization parameter p for a 4×4 block. Then the following process is defined in H.264 for the decoding,

$$\begin{aligned} \mathrm{T}^{-1}(\mathrm{Q}^{-1}(\boldsymbol{u})) &= \hat{\boldsymbol{w}}^{\mathrm{T}} \cdot (\boldsymbol{u} \cdot q[p]) \cdot \hat{\boldsymbol{w}}, \\ &= \boldsymbol{w}^{\mathrm{T}} \cdot ((\boldsymbol{u} \cdot h[p_{\text{rem}}] \cdot 2^{p_{\text{quo}}}) \otimes \boldsymbol{f}) \cdot \boldsymbol{w}, \\ &= \boldsymbol{w}^{\mathrm{T}} \cdot (\boldsymbol{u} \otimes (\boldsymbol{dq}[p_{\text{rem}}]) \cdot 2^{p_{\text{quo}}}) \cdot \boldsymbol{w} \cdot \frac{1}{64}, \end{aligned} \tag{2.5}$$

where $\boldsymbol{dq}[i] = (\boldsymbol{f} \cdot h[i] \cdot 64)$ with $6 > i \geq 0$ are constant matrices defined in the standard(see [63] for details). It is clear that the computation of (2.5) is then conducted using only integer operations.

Entropy Coding in H.264

H.264 supports two entropy coding methods for residual coding, i.e., context adaptive variable length coding (CAVLC) [3] in its baseline profile and context adaptive binary arithmetic coding (CABAC) [51] in its main profile.

Residual coding using CAVLC starts with the conventional run-length coding. CAVLC provides 7 tables for coding *level*s and a few tables for coding *run*s. A number of parameters are used to set up the table selection context, e.g., the total

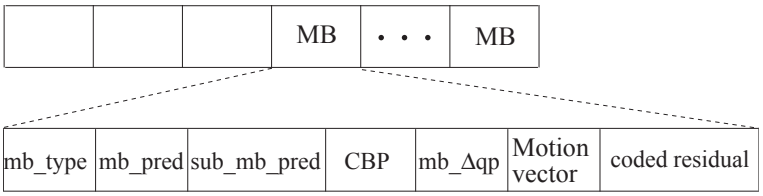


Figure 2.5: The syntax of macroblocks in H.264. The first three elements are used to signal the block partition and prediction mode. CBP stands for coded block pattern, which indicates which 8×8 blocks contain nonzero coefficients.

number of nonzero coefficients. More details will be introduced in Chapter 4 when a soft decision quantization algorithm is designed based on CAVLC. The Exp-Golomb codes are used for coding other syntax elements, e.g., the block mode, quantization step size, motion vectors, etc.

In order to achieve higher coding performance, H.264 supports binary arithmetic coding in its main profile. Arithmetic coding is generally superior to variable length coding because it can adapt to the symbol statistics and assign a non-integer number of bits to code a symbol. However, the complexity of arithmetic coding is high. In total, there are over 400 context models designed for various syntax elements. Details of CABAC is discussed in Chapter 5 before an SDQ algorithm is designed based on it.

The macroblock syntax for H.264 is summarized in Figure 2.5. As mentioned before, macroblock is the fundamental unit for mode selection. The coding process for a frame can be treated as repetitions of that for coding one macroblock. Thus, optimization for coding a frame can be broken down to optimization of individual macroblocks, which will simplify the implementation.

As discussed above, each individual coding part in H.264 has been well designed to achieve a good coding performance using the state-of-the-art technologies. Optimization of an individual part in H.264 alone will unlikely bring much improvement.

Meanwhile, a joint optimal design of the whole encoding structure is possible because the standard only specifies a syntax for the coded bit stream, leaving details of the encoding process open to a designer. In the next chapter, we will propose a rate distortion framework for jointly designing motion compensation, quantization and entropy coding in hybrid video coding, which is then applied to improve H.264 coding efficiency.

Chapter 3

An RD Optimization Framework for Hybrid Video Coding

In this chapter, we propose a joint design framework for hybrid video coding by optimizing trade-off between rate and distortion. Based on SDQ instead of conventional HDQ, the proposed framework allows us to jointly design motion compensation, quantization, and entropy coding by minimizing the actual RD cost. By the actual RD cost, we mean a cost based on the final reconstruction error and the entire coding rate. In the following, we first review RD optimization methods in the literature. Then, an SDQ scheme is introduced based on reviews of theoretical results on universal fixed-slope lossy coding. Based on the SDQ scheme, we formulate an RD optimization problem for hybrid video coding and then provide an iterative solution.

3.1 Related Rate Distortion Optimization Work

RD methods for video compression can be classified into two categories. The first category computes the theoretical RD function based on a given statistical model for video data, e.g., [11, 45]. In general, the challenge for designing a method in the first category is the model mismatch due to the non-stationary nature of video data

[38, 39]. The second category uses an operational RD function, which is computed based on the data to be compressed. This thesis is focused on developing operation RD methods.

Ramchandran et al. [61] developed an operational rate distortion framework for efficiently distributing bit budget among temporal and spatial coding methods for MPEG video compression. The rate distortion optimization problem was converted into a generalized bit allocation task. There was an issue of exponential complexity, which was tackled by utilizing a monotonicity property of operational rate distortion curves for dependent blocks/frames. The monotonicity property was constructed based on an assumption that rate distortion performance for coding one frame was monotonic in the effectiveness of prediction, which depended on the reproduction quality of reference frames. A pruning rule was then developed to reduce search complexity based on the monotonicity property. Generally speaking, the above assumption implies a linear relationship between distortion and residual coding rate. In fact, the above assumption is valid to a large extent. However, a problem here is that the total coding rate includes more than the rate for coding residuals. Motion vectors from motion compensation also need to be transmitted. For early standards such as MPEG-1, MPEG-2, motion compensation is based on a large block size of 16×16, leading to a small number of motion vectors to be transmitted. Motion vectors consume relatively few bits. It is then acceptable to apply the above assumption to simplify the rate distortion problem. However, when small block sizes are allowed for motion compensation such as 4×4 in H.264, motion vectors will consume a significant portion of the total coding bits. Consequently, it will not be able to find the optimal solution, either due to the approximation of the coding rate (when the monotonicity property is used) or because of the exponential complexity (when it is not used).

Using the generalized Lagrangian multiplier method [15], Wiegand *et al.* proposed a simple and effective operational RD method for motion estimation optimization [71, 75, 77]. The mode selection for motion estimation is conducted based

on the actual RD cost in a macroblock-by-macroblock manner[1]. For a given prediction mode, motion estimation is optimized based on an operational RD cost, which approximates the actual RD cost, as follows,

$$(f, \boldsymbol{v}) = \arg\min_{f,\boldsymbol{v}} d(\boldsymbol{x}, \boldsymbol{p}(m, f, \boldsymbol{v})) + \lambda \cdot (r(\boldsymbol{v}) + r(f)), \tag{3.1}$$

where $\boldsymbol{x}$ stands for the original image block, $\boldsymbol{p}(m, f, \boldsymbol{v})$ is the prediction with given prediction mode m, reference index f, and motion vector $\boldsymbol{v}$, $d(\cdot)$ is a distortion measure, $r(\boldsymbol{v})$ is the number of bits for coding $\boldsymbol{v}$, $r(f)$ is the number of bits for coding f, and λ is the Lagrangian multiplier.

Wen et. al [74] proposed an operational RD method for residual coding optimization in H.263+ using a trellis-based soft decision quantization design. In H.263+, residuals are coded with run-length codes followed by variable length coding (VLC). The VLC in H.263+ is simple and does not introduce any dependency among neighboring coefficients, while the dependency mainly comes from the run-length code. Therefore, a trellis structure is used to decouple the dependency so that a dynamic programming algorithm can be used to find the optimal path for quantization decisions. In the baseline of H.264, however, context adaptive VLC is used after the run-length coding. The context adaptivity introduces great dependency among neighboring coefficients, thus a new design criterion is needed to handle the context adaptivity for designing SDQ in H.264.

A recent study on soft decision quantization in [67] developed a linear model of inter-frame dependencies and a simplified rate model to formulate an optimization problem for computing the quantization outputs using a quadratic program. From the problem formulation point of view, our SDQ problem formulation shares the same spirit as that in [67], except that the latter one is more ambitious as it targets inter-frame dependencies. From the algorithm design point of view, [67] gives an optimized determination of transform coefficient levels by considering temporal de-

[1]RD optimization of mode selection for a group of macro-blocks in H.263 using dynamic programming was discussed in [76]. In H.264 reference software, however, the mode is independently selected for each macro-block [77].

pendencies, but neglecting other factors such as the specific entropy coding method, while the graph-based SDQ design to be presented latter in this thesis provides the optimal SDQ under certain conditions, i.e., motion prediction is given and CAVLC or CABAC is used for entropy coding.

Overall, there are two problems when designing an operational RD method. First, the formulated optimization problem is restricted and the RD cost is optimized only over motion estimation and quantization step sizes. Second, there is no simple way to solve the restricted optimization problem if the actual RD cost is used. Specifically, because of HDQ, there is no simple analytic formula to represent the actual RD cost as a function of motion estimation and quantization step sizes, and hence a brute force approach with high computational complexity is likely to be used to solve the restricted optimization problem [55]. For this reason, an approximate RD cost is often used in the restricted optimization problem in many operational RD methods. For example, the optimization of motion estimation in [77] is based on the prediction error instead of the actual distortion, which is the quantization error.

3.2 SDQ based on Fixed-Slope Lossy Coding

We now review a so-called fixed-slope lossy coding framework, based on which we propose a soft decision quantization scheme.

3.2.1 Overview of Fixed-Slope Lossy Coding

The framework for fixed-slope universal lossy data compression[2] has been well established in [27, 37, 36]. Consider a coding scheme shown in figure 3.1. The source $\mathbf{z}$ first passes through a mapping function $\alpha(\cdot)$. It is then encoded by a universal

[2]Related to fixed slope compression are entropy constrained [6] and conditional entropy constrained scalar/vector quantization. See [36, 10] for their difference and similarity.

lossless algorithm $\gamma(\cdot)$. To achieve rate reduction, $\alpha(\cdot)$ should be a multiple-to-one mapping. It is usually non-invertible, resulting in a distortion of $d(\mathbf{z}, \beta(\alpha(\mathbf{z})))$, where $d(\cdot)$ is the distortion measure. Define a length function $l_\gamma(\mathbf{u})$ for the lossless coding algorithm γ as the number of bits of the codeword that the algorithm assigns to $\mathbf{u}$. Then, the rate for coding $\mathbf{z}$ is computed as $r = l_\gamma(\alpha(\mathbf{z}))$. The problem of the fixed-slope lossy algorithm design is to find a solution of (α, β, γ) to minimize the actual rate distortion cost, i.e.,

$$\min_{\alpha,\beta,\gamma} \quad d(\mathbf{z}, \beta(\alpha(\mathbf{z}))) + \lambda \cdot l_\gamma(\alpha(\mathbf{z})), \tag{3.2}$$

where λ is a positive constant, which leads to the name of fixed-slope. As shown in Figure 3.2, a fixed-rate method finds a solution within the shadow area B, while the corresponding optimal solution approaches the crossing point of the RD curve and the line of $R = R_0$. A fixed-distortion method results in a solution within the shadow area C, while the corresponding optimal solution approaches the crossing point of the RD curve and the line of $D = D_0$. For a given λ, the fixed-slope method finds a solution within the shadow area A, which asymptotically approaches a point on the RD curve whose slope is $-\lambda$.

The theoretical basis of the fixed slope algorithm may be traced back to the variational descirption of Theorem 4.2.1 in [26] for evaluating rate-distortion functions. Denote $I(c)$ as the mutual information between a source P and the corresponding output by a channel with conditional pmf C, i.e., $I(q) = \sum_{j,k} p_j q_{k|j} \log \frac{q_{k|j}}{\sum_i p_i q_{k|j}}$. Suppose the distortion measure as $d(q) = \sum_{j,k} p_j q_{k|j} d(j,k)$. For a constant $\lambda > 0$, define a rate-distortion pair (R_λ, D_λ) parametrically by

$$\begin{aligned} R_\lambda &= \lambda D_\lambda + \min_c [I(q) + \lambda d(q)] \\ D_\lambda &= d(q) = \sum_j \sum_k p_j q^*_{k|j} d(j,k), \end{aligned} \tag{3.3}$$

where q^* is the conditional pmf yielding the minimum in (3.3). It has been proved that each value of the parameter λ will lead to a pair of (R_λ, D_λ), which is on the rate-distortion curve characterized as

$$R(D) = \min_{q \in \{d(q) < D\}} I(q), \tag{3.4}$$

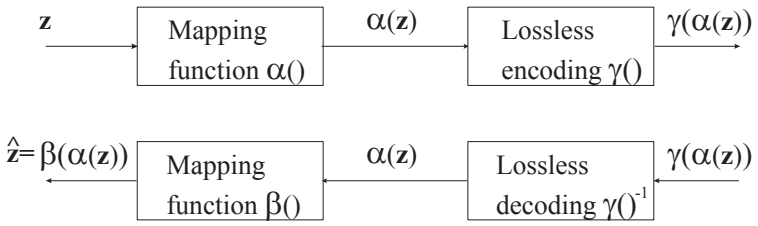


Figure 3.1: A universal lossy compression scheme. $\alpha(\cdot)$ is a non-invertible mapping function for encoding. $\beta(\cdot)$ is a mapping function for decoding. $\gamma(\cdot)$ is a lossless algorithm.

i.e.,

$$R_\lambda = R(D_\lambda).$$

The above result shows that for every given constant λ the unconstrained minimization in (3.3) will produce the optimal solution to the constrained optimization in (3.4. Note the difference between the contant λ in (3.3) and a Lagrange multiplier that may be used to solve the constrained minimization in (3.4) as

$$\min_{q,\lambda'}[I(q) + \lambda'(d(q) - D)], \tag{3.5}$$

where λ' is considered a variable. The difference between the minimization in (3.3) and (3.5) is that the former one results in a point on the rate-distortion curve with slope λ, while the latter results in a point with distortion D. Their results become the same if and only if D in (3.5) takes the value of D_λ in (3.3).

The fixed-slope method is generally superior to the fixed-rate or fixed-distortion method by its computational efficiency. Consider the asymptotic coding problem [47] based on the universal lossy scheme in Fig. 3.1. A fixed-rate method is equivalent to an inequality constrained problem

$$\min_{\alpha,\beta,\gamma} \; D(\mathbf{z}, \beta(\alpha(\mathbf{z}))), \text{ subject to } \gamma(\alpha(\mathbf{z})) - R_0 \leq 0,$$

which involves a search over $\{\alpha(\mathbf{z}) | \gamma(\alpha(\mathbf{z})) - R_0 \leq 0\}$ to minimize the distortion. Similarly, a fixed-distortion method is described as

$$\min_{\alpha,\beta,\gamma} \; \gamma(\alpha(\mathbf{z})), \text{ subject to } D(\mathbf{z}, \beta(\alpha(\mathbf{z}))) - D_0 \leq 0,$$

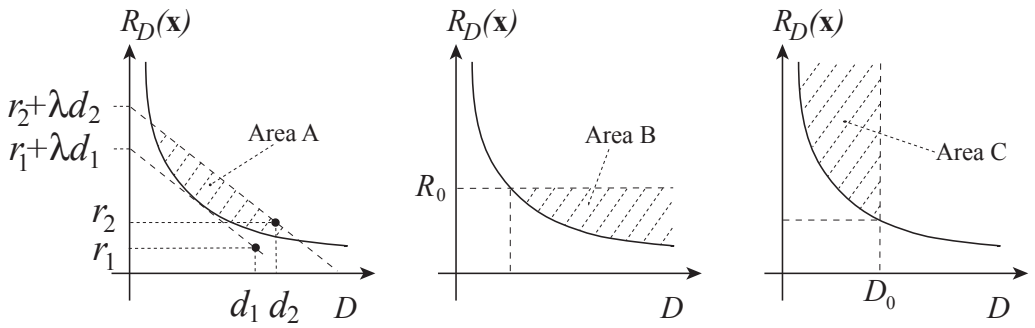


Figure 3.2: Illustrations of the rate distortion functions for coding schemes with fixed-slope, fixed rate, or fixed distortion. We use R and D to denote the rate and distortion, respectively.

which requires a search over $\{\alpha(\mathbf{z})|\delta(\mathbf{z},\beta(\alpha(\mathbf{z}))) - D_0 \leq 0\}$ to minimize the rate. The problem here is that both $\{\alpha(\mathbf{z})|\gamma(\alpha(\mathbf{z})) - R_0 \leq 0\}$ and $\{\alpha(\mathbf{z})|\delta(\mathbf{z},\beta(\alpha(\mathbf{z}))) - D_0 \leq 0\}$ have a very large size, which increases exponentially with the sequence dimension. Thus, they generally suffer from high coding complexity, although they possess asymptotic optimality [37, 47]. The fixed-slope method, however, is an unconstrained problem, for which there are some powerful methods to be used [2]. A successful example is the variable rate trellis source encoding algorithm in [36], where dynamic programming techniques are employed.

3.2.2 Soft Decision Quantization

Now, we investigate how the universal fixed-slope lossy coding scheme discussed above may be used to conduct SDQ in hybrid video coding optimization. Consider a 4×4 block, with quantized transform coefficient $\boldsymbol{u}$, prediction mode m, reference index f, motion vector $\boldsymbol{v}$, and quantization step size q. Its reconstruction is computed by

$$\hat{\boldsymbol{x}} = \boldsymbol{p}(m, f, \boldsymbol{v}) + \mathrm{T}^{-1}(\boldsymbol{u} \cdot q), \tag{3.6}$$

where $\boldsymbol{p}(m, f, \boldsymbol{v})$ is the prediction corresponding to $m, f, \boldsymbol{v}$ and $\mathrm{T}^{-1}(\cdot)$ is the inverse transform.

Conventionally, the constraint of (3.6) is used to derive a deterministic quantization procedure, i.e.,

$$\mathrm{HDQ}(\mathrm{T}(\boldsymbol{z})) = \mathrm{round}([\mathrm{T}(\boldsymbol{z}) + \delta \cdot q]/q), \tag{3.7}$$

which mainly minimizes the quantization distortion $d(\boldsymbol{x}, \hat{\boldsymbol{x}})$, where $\boldsymbol{z} = \boldsymbol{x} - \boldsymbol{p}(m, f, \boldsymbol{v})$. The factor δ is an offset parameter for adapting the quantization outputs to the source distribution to some extend. There are empirical studies on determining δ according to the signal statistics to improve the RD compression efficiency[79]. Still, this is an HDQ process. From the syntax-constrained optimization point of view, however, there is no deterministic relationship such as (3.7) between quantization outputs and $(m, f, \boldsymbol{v})$. Examine the fixed-slope lossy scheme of (3.2) under the circumstance of optimizing H.264 baseline coding. In case of H.264-compliant coding, the decoding mapping $\beta(\cdot)$ and the lossless coding algorithms γ and γ^{-1} have been specified in the standard, i.e., γ and γ^{-1} accord to entropy encoding and decoding in H.264, respectively, and $\beta(\cdot) = \mathrm{T}^{-1}(\mathrm{Q}^{-1}(\cdot))$, where $\mathrm{T}^{-1}(\cdot)$ and $\mathrm{Q}^{-1}(\cdot)$ are the inverse DCT and de-quantization, respectively. In case that HDQ was used, we would have $\alpha(\cdot) = \beta^{-1}(\cdot) = \mathrm{Q}(\mathrm{T}(\cdot))$. However, the relationship of $\alpha(\cdot) = \beta^{-1}(\cdot)$, though appearing to be true, finds no ground in the standard specification[3]. Instead, the H.264 standard only specifies $\beta(\cdot)$ and $\gamma(\cdot)$, leaving $\alpha(\boldsymbol{z})$ a free parameter for minimizing the RD cost. In this case, the problem of (3.2) reduces to a search for $\boldsymbol{u} = \alpha(\boldsymbol{z})$ to minimize the RD cost, i.e.,

$$\boldsymbol{u} = \arg\min_{\boldsymbol{u}} \quad d(\boldsymbol{z}, \mathrm{T}^{-1}(\boldsymbol{u} \cdot q)) + \lambda \cdot r_{\gamma}(\boldsymbol{u}). \tag{3.8}$$

The minimization in (3.8) is over all possible quantized values. In general, such a $\boldsymbol{u}$ will not be obtained by the hard decision process via (3.7), and the quantization by (3.8) is called SDQ [31].

Here is a simple example illustrating the idea underlying the SDQ. Consider a quantization step size $q = 5$, a block of transform coefficients

$$\boldsymbol{c} = \mathrm{T}(\boldsymbol{x} - \boldsymbol{p}(m, f, \boldsymbol{v})) = (84, 0, -8, 17, 0, -11, -8, 1),$$

[3]Actually, this is true for most recent video compression standards, as they only specify the decoding syntax and leave the encoding open for optimization.

and the entropy coding method of context adaptive variable length coding (CAVLC) in H.264. (See Section 2.3.2 for review of hybrid coding in H.264.) The quantization output given by conventional HDQ is

$$\boldsymbol{u}'=(17,\ 0,\ -2,\ 3,\ 0,\ -2,\ -2,\ 0).$$

In this case, the resulting distortion for coding the block is 15, and the number of bits resulting from using CAVLC to code $\boldsymbol{u}'$ is 45. On the other hand, with $\lambda = 30$, an SDQ method may output,

$$\boldsymbol{u}=(17,\ 0,\ -2,\ 4,\ 0,\ -2,\ -1,\ 0).$$

In this case, the resulting distortion is 25, but the number of bits needed for CAVLC to code $\boldsymbol{u}$ reduces to 27. With $\lambda = 30$, the RD costs resulting from $\boldsymbol{u}'$ and $\boldsymbol{u}$ are respectively 1365 and 835 with the latter significantly smaller than the former. Note that the value -8 is quantized into both -2 and -1 in $\boldsymbol{u}$, as

$$c_3{=}-8,\ u_3{=}-2 \text{ and } c_7{=}-8,\ u_7{=}-1.$$

Clearly, SDQ can trade off a little more distortion for a significant rate reduction for using CAVLC.

The idea of trading off a little distortion for a better RD performance has already been utilized partially in the H.264 reference software, however, in an ad hoc way[79]. A whole block of quantized coefficients is discarded under certain conditions, e.g., when there is only one non-zero coefficient taking a value of 1 or -1. This is equivalent to quantizing that coefficient to 0, although a hard decision scalar quantizer would output 1 or -1 for that coefficient. Such simple practice has been well justified by experimental results [79]. To get better compression performance, it is interesting and desirable to conduct SDQ in a systematic way of (3.8).

Overall, the SDQ scheme is inspired by the fixed-slope universal lossy data compression scheme considered in [37], which was first initiated in [29] and was latter

extended in [36]. Other related works on practical SDQ include without limitation SDQ in JPEG image coding and H.263+ video coding (see [62, 10, 28, 74, 67] and references therein). In [62, 10], partial SDQ called rate-distortion optimal thresholding was considered. Recently, Yang and Wang [28] successfully developed an algorithm for optimal SDQ in JPEG image coding to further improve the compression performance of a standard JPEG image codec. Without considering optimization over motion estimation and quantization step sizes, Wen et. al [74] proposed a trellis-based algorithm for optimal SDQ in H.263+ video coding, which, however, is not applicable to SDQ design in H.264 due to the inherent difference in the entropy coding stages of H.264 and H.263+. In [67], Schumitsch et. al. studied inter-frame optimization of transform coefficient levels based on a simplified linear model of inter-frame dependencies. Although the SDQ principle is not new and this thesis is not the first attempt [28] to apply SDQ to practical coding standards either, designing algorithms for optimal or near optimal SDQ in conjunction with a specific entropy coding method is still quite challenging, especially when the involved entropy coding method is complicated. Different entropy coding methods require different algorithms for SDQ. In some cases, for example, SDQ for GIF/PNG coding where the entropy coding methods are the Lempel-Ziv[84][85] algorithms, the SDQ design problem is still open [35]. Fortunately, in the case of H.264, we are able to tackle the SDQ design issues associated with CAVLC and CABAC in H.264 [32, 30, 33]. Furthermore, our studies in SDQ within the fixed slope scheme constitutionally leads to a new framework for jointly designing motion prediction, quantization, and entropy coding in hybrid video coding, as described in the next section.

3.3 Rate Distortion Optimization Framework for Hybrid Video Coding

Based on the SDQ scheme of (3.8), we now examine the maximal variability and flexibility an hybrid encoder can enjoy when decoding syntaxes are given. Then, an RD optimization problem is formulated for minimizing the actual RD cost and an iterative solution is developed after.

3.3.1 Optimization Problem Formulation

A conventional RD optimization framework for hybrid video coding is based on HDQ of (3.7). Consider RD optimization for a whole frame $\boldsymbol{X}$, which consists of a group of blocks. Denote prediction modes, reference frames, motion vectors, and quantization step sizes as $\boldsymbol{m}$, $\boldsymbol{f}$, $\boldsymbol{V}$, and $\boldsymbol{q}$, respectively. The actual RD cost is

$$\begin{aligned} &J_{\boldsymbol{X}}(\boldsymbol{m}, \boldsymbol{f}, \boldsymbol{V}, \boldsymbol{q}) \\ &\quad = ||\boldsymbol{Z} - \hat{\boldsymbol{Z}}||^2 + \lambda \cdot [r(\boldsymbol{m}) + r(\boldsymbol{f}) + r(\boldsymbol{V}) + r(\boldsymbol{q}) + r(\text{ HDQ}[\text{T}(\boldsymbol{Z})]\ )\ ], \end{aligned} \tag{3.9}$$

where $\boldsymbol{Z} = \boldsymbol{X} - \boldsymbol{P}(\boldsymbol{f}, \boldsymbol{m}, \boldsymbol{V})$, $\boldsymbol{P}(\boldsymbol{f}, \boldsymbol{m}, \boldsymbol{V})$ is the prediction, $\hat{\boldsymbol{Z}}$ is the residual reconstructed from the hard decision quantization outputs HDQ(T($\boldsymbol{Z}$)), and λ is a constant. The conventional RD optimization framework for hybrid video compression can then be summarized as follows,

$$\min_{\boldsymbol{m}, \boldsymbol{f}, \boldsymbol{V}, \boldsymbol{q}} \ J_{\boldsymbol{X}}(\boldsymbol{m}, \boldsymbol{f}, \boldsymbol{V}, \boldsymbol{q}). \tag{3.10}$$

However, it is easy to see that HDQ is not desirable for minimizing the RD cost because with HDQ the minimizing of the actual RD cost is impractical, i.e., it requires to go through the residual coding procedure for many time [55]. Moreover, HDQ is not required by any hybrid coding standard. Indeed, inspired by the SDQ scheme of (3.8), we discover that given motion prediction and a quantization step size, the quantization output itself is a free parameter and one has the flexibility to choose the desired quantization output in order to optimize trade-off between rate

and distortion rather than to minimize the distortion only [34] [32], as discussed in section 3.2.

Using SDQ instead of conventional HDQ, an optimization problem for jointly designing motion prediction, quantization, and entropy coding in a hybrid coding structure is formulated as follows,

$$\min_{\boldsymbol{m},\boldsymbol{f},\boldsymbol{V},\boldsymbol{q},\boldsymbol{U}} J_{\boldsymbol{X}}(\boldsymbol{m},\boldsymbol{f},\boldsymbol{V},\boldsymbol{q},\boldsymbol{U}), \tag{3.11}$$

where

$$J_{\boldsymbol{X}}(\boldsymbol{m},\boldsymbol{f},\boldsymbol{V},\boldsymbol{q},\boldsymbol{U}) = ||\boldsymbol{Z} - \mathrm{T}^{-1}(\mathrm{Q}^{-1}(\boldsymbol{U}))||^2 + \lambda\cdot[r(\boldsymbol{m})+r(\boldsymbol{f})+r(\boldsymbol{V})+r(\boldsymbol{q})+r(\boldsymbol{U})] .$$

A simple comparison between the proposed framework in (3.11) and the conventional one in (3.10) reveals an advantage of the proposed framework, i.e.,

$$\min_{\boldsymbol{m},\boldsymbol{f},\boldsymbol{V},\boldsymbol{q},\boldsymbol{U}} J_{\boldsymbol{X}}(\boldsymbol{m},\boldsymbol{f},\boldsymbol{V},\boldsymbol{q},\boldsymbol{U}) \leq \min_{\boldsymbol{m},\boldsymbol{f},\boldsymbol{V},\boldsymbol{q}} J_{\boldsymbol{X}}(\boldsymbol{m},\boldsymbol{f},\boldsymbol{V},\boldsymbol{q}),$$

since for any given $\boldsymbol{m},\boldsymbol{f},\boldsymbol{V},\boldsymbol{q}$, we can always apply SDQ in (3.8) to reduce the RD cost in (3.9). Furthermore, the problem of optimizing the actual RD cost becomes tractable algorithmically by (3.11), i.e., as discussed in Section 3.3.2, an iterative solution is easily established to optimize over $\boldsymbol{m}$, $\boldsymbol{f}$, $\boldsymbol{V}$, $\boldsymbol{q}$ and $\boldsymbol{U}$. The solution is at least feasible, although it may not be proved to be globally optimal. On the other hand, with the conventional framework of (3.10), it is impractical to optimize the actual RD cost over $\boldsymbol{f}$, $\boldsymbol{V}$, and $\boldsymbol{q}$, because it would require to go through the residual coding procedure to evaluate the cost for all possible $\boldsymbol{f}$, $\boldsymbol{V}$, and $\boldsymbol{q}$. Overall, due to SDQ, the new framework supports a better RD performance and features a feasible solution to minimizing the actual RD cost for hybrid video coding.

3.3.2 Problem Solution

In general, (3.11) is difficult to solve due to the mutual dependency among $\boldsymbol{m}$, $\boldsymbol{f}$, $\boldsymbol{V}$, $\boldsymbol{q}$, $\boldsymbol{U}$. To make the problem tractable, we propose an iterative solution, in which motion estimation and residual coding are optimized alternately. Specifically, three RD optimization algorithms are developed—one for SDQ given motion

estimation and quantization step sizes, one for optimizing residual coding given motion estimation, and one for overall optimization of hybrid video encoding for each individual frame with given reference frames—with them embedded in the indicated order.

Optimal Soft Decision Quantization

Given $(\boldsymbol{m}, \boldsymbol{f}, \boldsymbol{V}, \boldsymbol{q})$, the minimization problem of (3.11) becomes

$$\min_{\boldsymbol{U}} ||\boldsymbol{Z} - \mathrm{T}^{-1}(\mathrm{Q}^{-1}(\boldsymbol{U}))||^2 + \lambda \cdot r(\boldsymbol{U}), \tag{3.12}$$

where $\boldsymbol{Z}$ is the residual corresponding to given $(\boldsymbol{m}, \boldsymbol{f}, \boldsymbol{V}, \boldsymbol{q})$. It is easy to see that the exact optimal SDQ solution to (3.12) depends on entropy coding, which determines the rate function $r(\cdot)$. Furthermore, the entropy coding method is application-dependent. Different applications have different entropy coding methods and hence different SDQ solutions. Some early work on practical (optimal or suboptimal) SDQ includes without limitation SDQ in JPEG image coding and H.263+ video coding (see [67, 10, 28, 74, 62] and references therein). In this thesis, we focus on RD optimization with H.264 compatibility. Since H.264 supports two entropy coding methods, i.e., CAVLC and CABAC, two SDQ algorithms are to be developed. Specifically, a graph-based SDQ design based on CAVLC is to be presented in Chapter 4 when the proposed RD framework is applied to optimize H.264 baseline profile encoding. Another SDQ design based on CABAC is to be presented in Chapter 5 when the proposed RD framework is applied to optimize H.264 main profile encoding.

Residual Coding Optimization

Residual coding optimization here refers to a partial solution of (3.11) with given $(\boldsymbol{m}, \boldsymbol{f}, \boldsymbol{V})$. Essentially, it involves alternating optimization over $\boldsymbol{U}$ with given $\boldsymbol{q}$, which is SDQ, and optimization over $\boldsymbol{q}$ with given $\boldsymbol{U}$. Specifically, given $(\boldsymbol{m}, \boldsymbol{f}, \boldsymbol{V})$,

in residual coding optimization, we compute

$$\arg\min_{\boldsymbol{q},\boldsymbol{U}} ||\boldsymbol{Z} - \mathrm{T}^{-1}(\mathrm{Q}^{-1}(\boldsymbol{U}))||^2 + \lambda \cdot (r(\boldsymbol{q}) + r(\boldsymbol{U})). \tag{3.13}$$

In general, algorithms for solving (3.13) are to be designed based on specific coding syntaxes of $\mathrm{T}^{-1}(\mathrm{Q}^{-1}(\cdot))$ and $r(\cdot)$. As discussed above, the SDQ design is closely related to a given entropy coding method. However, when the SDQ design is given, it is easy to obtain a solution to (3.13). In the following, we present our solution to (3.13), which is developed for optimizing H.264-compatible coding.

Examining the distortion term in (3.13), we see that its computation is macro-block wise additive. As to be discussed later, even though the term $r(\boldsymbol{U})$ is not strictly macroblock-wise additive, the adjacent block dependency used in coding $\boldsymbol{U}$ is so weak that we can ignore it in our optimization and simply regard $r(\boldsymbol{U})$ as being block-wise additive. Thus, the main difficulty lies in the term of $r(\boldsymbol{q})$, which represents a first order predictive coding method [79] in H.264. As such, the optimization problem in (3.13) for H.264 can not be solved in a macroblock-by-macroblock manner.

To tackle the adjacent macro-block dependency from $r(\boldsymbol{q})$, we develop a trellis structure with K stages and 52 states at each stage (H.264 specifies 52 quantization step sizes). Each stage accords to a macro-block, while each state accords to a quantization step size. States between two neighboring stages are fully connected with each other. The RD cost for a transition between the ith state at the $(m-1)$th stage to the jth state at the mth stage can be computed by two parts, i.e., the coding rate of $r(q_j - q_i)$ and the RD cost for coding the mth macro-block using q_j, which is computed using SDQ. The RD cost for each state j at the initial stage is equal to the RD cost resulting from encoding the first macro-block using q_j and the corresponding optimal SDQ. Then, dynamic programming can be used to solve (3.13) completely.

Apparently, the above solution is computationally expensive as it involves running SDQ for each one of 52 states at each stage and then searching the whole

trellis. In practice, however, there is no need for this full scale dynamic programming because the RD cost corresponding to the quantization output $\boldsymbol{U}$ is much greater than that corresponding to the quantization step size $\boldsymbol{q}$ [4]. This implies that very likely, the globally optimal quantization step size for each macro-block will be within a small neighboring region around the best quantization step size obtained when $r(\boldsymbol{q})$ is ignored in the cost and one can apply dynamic programming to a much reduced trellis with states at each stage limited only to such a small neighborhood. To this end, we propose the following procedure to find the best $\boldsymbol{q}$ when $r(\boldsymbol{q})$ is ignored.

Step 1 : For a given block[5], initialize q using the following empirical equation proposed in [75] with a given λ in conjunction with (2.4):

$$\lambda = 0.85 \cdot 2^{(p-12)/3}. \tag{3.14}$$

Step 2 : Compute $\boldsymbol{u}$ by the SDQ algorithm[6].

Step 3 : Fix $\boldsymbol{u}$. Compute q by solving $\frac{\partial J}{\partial q} = 0$. By ignoring $r(\boldsymbol{q})$, we have $q = |\mathrm{T}(\boldsymbol{z}) \cdot \boldsymbol{u}/(\boldsymbol{u} \cdot \boldsymbol{u})|$, which is then rounded to one of the 52 predefined values in H.264.

Step 4 : Repeat Steps 2 and 3 until the decrement of the RD cost is less than a prescribed threshold.

[4]One quantization step size is used for a whole 32×32 macroblock, which accords to 32×32 quantization outputs. Besides, the dynamic range of quantization step size is [1, 52] while the dynamic range of a quantization output is [0, 255].

[5]As we use $\boldsymbol{U}$, $\boldsymbol{q}$ to represent quantization outputs and quantization step sizes for a whole frame, we use $\boldsymbol{u}$, q to represent those for any macro-block, with subscript omitted for simplicity.

[6] We assume an SDQ is given while discussing the residual coding optimization algorithm in this section. For applying the proposed framework, an SDQ is firstly developed based on a specific entropy coding method. Then, the SDQ is embedded to the residual coding optimization algorithm, which is further embedded into the overall optimization algorithm presented in the next section.

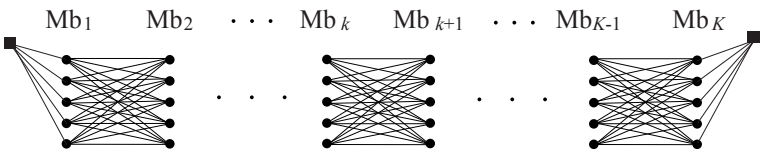


Figure 3.3: A reduced trellis structure for the residual coding optimization. The five states accord to quantization step sizes of $q_k(p_k-2), q_k(p_k-1), q_k(p_k), q_k(p_k+1), q_k(p_k+2)$.

Simulations show that (3.14) makes a good initial point. After one iteration, the obtained q is quite close to the best quantization step size with $r(\boldsymbol{q})$ being ignored. We then select a neighboring region of $[q-2, q+2]$ to build up the trellis at stage k, as shown in Figure 3.3, and hence the computation complexity is greatly reduced.

Experiments have been conducted to compare the performance of the reduced trellis structure with that of the full trellis structure. Specifically, we encode 20 frames with 10 frames from "foreman.qcif" and 10 frames from "carphone.qcif". In total, there are 1980 macroblocks. We compare the optimal quantization step sizes obtained using the full trellis and those obtained using the reduced trellis. There is only one macroblock in "carphone.qcif" that chooses a different quantization step size by using the full trellis and the reduced trellis. Correspondingly, the rate distortion performance are shown in Table 3.1. It is shown that dynamic programming applied to this reduced trellis achieves almost the same performance as that applied to the full trellis.

Table 3.1: RD performance for using the full trellis and the reduced trellis.

	Reduced trellis	Full trellis
PSNR (dB)	36.885	36.886
Number of bits	19876	19878

The joint optimization algorithm

Based on the algorithm for the near optimal residual coding, a joint optimization algorithm for solving (3.11) is proposed to alternately optimize motion estimation and residual coding as follows.

Step 1 : (Motion estimation) For given residual reconstruction $\hat{\boldsymbol{Z}}(\boldsymbol{q}, \boldsymbol{U})$, we compute $(\boldsymbol{m}, \boldsymbol{f}, \boldsymbol{V})$ by,

$$\min_{\boldsymbol{m},\boldsymbol{f},\boldsymbol{V}} d(\boldsymbol{X} - \boldsymbol{P}(\boldsymbol{m}, \boldsymbol{f}, \boldsymbol{V}), \hat{\boldsymbol{Z}}) + \lambda \cdot (r(\boldsymbol{m}) + r(\boldsymbol{f}) + r(\boldsymbol{V})). \tag{3.15}$$

which is equivalent to (3.11) for given $(\boldsymbol{q}, \boldsymbol{U})$.

Step 2 : (Residual coding) For given $(\boldsymbol{m}, \boldsymbol{f}, \boldsymbol{V})$, the process in Section 3.3.2 is used to find $(\boldsymbol{q}, \boldsymbol{U})$.

Step 3 : Repeat Steps 1 and 2 until the decrement of the actual RD cost is less than a given threshold.

We now study the solution to (3.15), which involves mode selection and motion estimation. In [77], the prediction mode is selected for each macroblock by computing the actual RD cost corresponding to each mode and choosing the one with the minimum. This method of mode selection is also used here. For a pixel block $\boldsymbol{x}$ with its residual reconstruction $\hat{\boldsymbol{z}}$ and a given mode m, $(f, \boldsymbol{v})$ is computed by

$$(f, \boldsymbol{v}) = \arg\min_{f,\boldsymbol{v}} \; d(\; \boldsymbol{x} - \hat{\boldsymbol{z}},\; \boldsymbol{p}(m, f, \boldsymbol{v})\;) + \lambda \cdot (r(\boldsymbol{v}) + r(f)). \tag{3.16}$$

Compare (3.16) with (3.1). For given $\hat{\boldsymbol{z}}$, (3.16) is equivalent to searching for a prediction to match $\boldsymbol{x} - \hat{\boldsymbol{z}}$ in (3.1). Thus, the same search algorithm is used to solve (3.16) as the one for (3.1) in [77]. The computational complexity for (3.16) and (3.1) is almost the same since the time for computing $\boldsymbol{x} - \hat{\boldsymbol{z}}$ is ignorable.

By its iterative nature, the above joint optimization algorithm is not guaranteed to converge to the global optimal solution of (3.11). However, it does converge in the sense that the actual rate distortion cost is decreasing at each iteration

step. The computational complexity of the proposed iterative algorithm comes from three parts, i.e., optimal residual coding, motion vector computation, and mode selection. In case of H.264, the motion vector updating part and the mode selection part hardly cause any increase in the computational complexity compared to the rate distortion optimization method in [77], as discussed in the above. The main extra computational complexity results from the optimal residual coding part, particularly the SDQ algorithm.

3.4 Chapter Summary

In this chapter, we have discussed RD optimization for hybrid video compression. Inspired by the universal fixed-slope lossy coding scheme, we have discovered a new free parameter for RD optimization, i.e., the quantization output, based on which an RD optimization framework is proposed to minimize the actual RD cost. Within the framework, we have proposed three algorithms—SDQ, residual coding optimization, and an iterative overall algorithm—with them embedded in the indicated order. In general, the proposed framework may be applied to optimize RD trade-off for any hybrid coding scheme by developing three algorithms according to corresponding coding syntaxes. In this chapter, we have presented details of the residual coding algorithm and the joint optimization algorithm for optimizing RD trade-off in H.264. The real challenge for algorithm design, actually, comes from the SDQ design. In the following chapters, we will discuss detailed SDQ designs and their applications in the proposed RD framework to optimizing H.264 baseline profile encoding and main profile encoding, respectively.

Chapter 4

RD Optimal Coding with H.264 Baseline Compatibility

In this chapter, we apply the proposed RD framework to optimizing H.264 baseline profile encoding. As discussed in Chapter 3, a key step in the application of the proposed framework is to design SDQ in conjunction with a specific entropy coding method, i.e., CAVLC in H.264 baseline profile in this case. Once an SDQ algorithm is designed, it can be called as a subroutine by the residual coding optimization algorithm, which is then called by the overall algorithm, as discussed in Chapter 3.

4.1 Review of CAVLC

CAVLC is used to code zig-zag ordered quantized transform coefficients[1] in the H.264 baseline profile. For a given zig-zag sequence $\bar{\boldsymbol{u}}$, CAVLC encoding is conducted in the reverse order. In particular, the CAVLC encoding algorithm is summarized as follows [3]:

1. Initialization. The sequence is scanned in the reverse order to initialize two sets of parameters. The first set includes *TotalCoefficients* (refereed to as *TC*

[1]Quantized transform coefficients are also referred to as transform coefficient levels.

hereafter), *T1s*, and *TotalZeros*, which represent the total number of non-zero coefficients, the number of trailing coefficients with value ± 1, and the number of zero coefficients between the first non-zero coefficient and the scan end, respectively. The definition of *T1s* is based on an observation that the highest frequency non-zero coefficients in the scan are often a sequence of ± 1, called trailing ones. CAVLC allows at most 3 trailing ones to be specially handled, i.e., $T1s \leq 3$. The second set is a series of $(run, level)$ pairs, where *level* means a non-zero coefficient and *run* is the number of zeros between the current *level* and the next *level*.

2. Encoding *CoeffToken*. *TC* and *T1s* are combined into one parameter, i.e., *CoeffToken*, to be encoded. Four look-up tables are defined for encoding *CoeffToken*. The selection depends on the numbers of non-zero coefficients in upper and left-hand previously coded blocks, N_u and N_l. Specifically, a table is selected based on $M = (N_u + N_l)/2$ by the following procedure:

```
if(0<=M<2) use table Num-V0          ;
if(2<=M<4) use table Num-V1          ;
if(4<=M<8) use table Num-V2          ;
if(M>=8)   6-bit fixed length code   ;
```

3. Encoding the sign of each trailing one. One bit is used to signal the sign, i.e., 0 for + and 1 for -. Note that the number of trailing ones has already been transmitted.

4. Encoding *level*s. 7 VLC tables, named as Vlc(i) with $0 \leq i \leq 6$, are used to encode all *level*s other than trailing ones. The table selection criteria are summarized in the following pseudo codes.

```
// Choose a table for the first level
if(TotalCoeffs>10 && T1s<3)   i = 1   ; // use Vlc(1)
```

```
else                                i = 0   ; // use Vlc(0)
// Update the table selection after coding each level
vlc_inc[7] = {0, 3, 6, 12, 24, 48, 65536}   ;
if(level>vlc_inc[i])                i ++          ;
if(level>3 && (Be the first non-1 level))     i = 2 ;
```

5. Encoding *TotalZeros*. One out of 15 tables is chosen based on *TC* to encode *TotalZeros*.

6. Encoding *run*s. For each *run*, a parameter called *ZerosLeft* (referred to as *ZL* hereafter) is defined as the number of zeros between the current *level* and the scan end. It is then used to select one table out of 7 to encode the current *run*. E.g., *ZL* equals to *TotalZeros* for the first *run*.

In summary, CAVLC is designed to take advantage of some empirical observations on quantized coefficients. First, they are commonly sparse, i.e., containing mostly zeros. Run-length coding is used to represent consecutive zeros efficiently. Second, it is very likely that the trailing nonzero coefficients after the zig-zag scan take values of ± 1. The trailing one rule is specially developed to handle these *level*s. Third, the magnitude of non-zero coefficients tends to be higher at the start of the zig-zag scan and lower towards the higher frequencies. The *level* coding tables Vlc(0-6) are constructed according to this tendency. All these delicate designs together pave the way for CAVLC to be adopted in H.264.

4.2 SDQ Design based on CAVLC

In this section, we present a graph-based SDQ algorithm based on CAVLC, which solves the SDQ problem of (3.12). Clearly, for given residual and $\boldsymbol{q}$, the distortion term in (3.12) is block-wise additive. Note that $\boldsymbol{U} = \{\boldsymbol{u}_1, \cdots, \boldsymbol{u}_{16K}\}$, where K is the number of macro-blocks in a frame and $16K$ is the number of 4×4 blocks

there. In H.264, encoding of each block $\boldsymbol{u}_k$ depends not only on $\boldsymbol{u}_k$ itself, but also on its two neighboring blocks. However, such dependency is very weak and the number of bits needed to encode $\boldsymbol{u}_k$ largely depends on $\boldsymbol{u}_k$ itself. Therefore, in the optimization problem given in (3.12) for the whole frame, we will decouple such weak dependency. In doing so, the optimization of the whole frame can be solved in a block by block manner with each block being 4×4. That is, the optimal $\boldsymbol{U}$ can be determined independently for each $\boldsymbol{u}_k$. By omitting the subscript, the optimization problem given in (3.12) now reduces to,

$$\boldsymbol{u} = \arg\min_{\boldsymbol{u}} \; d(\boldsymbol{z}, \mathrm{T}^{-1}(\boldsymbol{u} \cdot q)) + \lambda \cdot r(\boldsymbol{u}), \tag{4.1}$$

where $r(\boldsymbol{u})$ is the number of bits needed for encoding $\boldsymbol{u}$ using CAVLC given that its two neighboring blocks have been optimized.

In general, SDQ is a search in a vector space of quantization outputs for trade-off between rate and distortion. The efficiency of the search largely depends on how we may discover and utilize the structure of the vector space, which features the de-quantization syntax and the entropy coding method of CAVLC. In this study, we propose to use a dynamic programming technique to do the search, which requires an additive evaluation of the RD cost. In the following, we first show the additive distortion computation in the DCT domain based on the de-quantization syntax in H.264 as reviewed in Section 2.3.2. Second, we design a graph for additive evaluation of the rate based on analysis of CAVLC, with states being defined according to *level* coding rules and connections being specified according to *run* coding rules. Finally, we discuss the optimality of the graph-based algorithm, showing that the graph design helps to solve the minimization problem of (4.1).

4.2.1 Distortion Computation in the DCT domain

The distortion term in (4.1) is defined in the pixel domain. It contains inverse DCT, which is not only time consuming, but also makes the optimization problem intractable. Consider that DCT is a unitary transform, which maintains the Eu-

clidean distance. We choose the Euclidean distance for $d(\cdot)$. Then, the distortion can be computed in the transform domain in an element-wise additive manner.

As reviewed in Section 2.3.2, the transform and quantization in H.264 are combined together. Specifically, the residual reconstruction process is

$$\mathrm{T}^{-1}(\boldsymbol{u} \cdot q) = \boldsymbol{w}^T \cdot (\boldsymbol{u} \otimes \boldsymbol{dq}[p_{\mathrm{rem}}] \cdot 2^{p_{\mathrm{quo}}}/64) \cdot \boldsymbol{w}. \tag{4.2}$$

Since $\hat{\boldsymbol{w}}$ defines a unitary transform, we have

$$||\hat{\boldsymbol{w}}^{\mathrm{T}} \cdot \boldsymbol{Y} \cdot \hat{\boldsymbol{w}}||^2 = ||\boldsymbol{Y}||^2.$$

Equivalently, that is,

$$||\boldsymbol{w}^{\mathrm{T}} \cdot \boldsymbol{Y} \cdot \boldsymbol{w}||^2 = ||\boldsymbol{Y} \otimes \boldsymbol{B}||^2, \tag{4.3}$$

where $\boldsymbol{Y}$ is any 4×4 matrix, and

$$\boldsymbol{B} = \begin{pmatrix} 4 & \sqrt{10} & 4 & \sqrt{10} \\ \sqrt{10} & \frac{5}{2} & \sqrt{10} & \frac{5}{2} \\ 4 & \sqrt{10} & 4 & \sqrt{10} \\ \sqrt{10} & \frac{5}{2} & \sqrt{10} & \frac{5}{2} \end{pmatrix}.$$

Note that $\boldsymbol{B}$ is obtained based on the given matrixes of $\boldsymbol{w}$ and $\hat{\boldsymbol{w}}$ as shown in Section 2.3.2. Consider $\boldsymbol{z} = \hat{\boldsymbol{w}}^{\mathrm{T}}(\hat{\boldsymbol{w}} \cdot \boldsymbol{z} \cdot \hat{\boldsymbol{w}}^{\mathrm{T}})\hat{\boldsymbol{w}}$. Applying (4.3), we compute the the distortion term in (4.1) with the Euclidean measure by

$$\begin{aligned} D &= ||\boldsymbol{z} - \boldsymbol{w}^{\mathrm{T}} \cdot (\boldsymbol{u} \otimes \boldsymbol{dq}[p_{\mathrm{rem}}] \cdot 2^{p_{\mathrm{quo}}}/64) \cdot \boldsymbol{w}||^2 \\ &= ||\boldsymbol{w}^{\mathrm{T}} \cdot \Big((\hat{\boldsymbol{w}} \cdot \boldsymbol{z} \cdot \hat{\boldsymbol{w}}^{\mathrm{T}}) \otimes \boldsymbol{f} - \boldsymbol{u} \otimes \boldsymbol{dq}[p_{\mathrm{rem}}] \cdot 2^{p_{\mathrm{quo}}}/64\Big) \cdot \boldsymbol{w}||^2 \\ &= ||\boldsymbol{c} - \boldsymbol{u} \otimes \boldsymbol{dq}[p_{\mathrm{rem}}] \cdot 2^{p_{\mathrm{quo}}}/64 \otimes \boldsymbol{B}||^2 \end{aligned} \tag{4.4}$$

where $\boldsymbol{c} = (\boldsymbol{w} \cdot \boldsymbol{z} \cdot \boldsymbol{w}^{\mathrm{T}}) \otimes \boldsymbol{f}$. The equation of (4.4) brings to us two advantages. The first is the high efficiency for computing distortion. Note that $\boldsymbol{B}$ and $\boldsymbol{dq}$ are constant matrices defined in the standard. $\boldsymbol{c}$ is computed before soft decision quantization for given $\boldsymbol{z}$. Thus, the evaluation of D consumes only two integer multiplications together with some shifts and additions per coefficient. More importantly, the second advantage is the resulted element-wise additive computation

of distortion, which enables us to solve the soft decision quantization problem using the Viterbi algorithm to be presented later.

After applying the result of (4.4) to (4.1), the soft decision quantization problem becomes

$$\boldsymbol{u} = \arg\min_{\boldsymbol{u}} ||\boldsymbol{c} - \boldsymbol{u} \otimes \boldsymbol{dq}[p_{\text{rem}}] \cdot 2^{p_{\text{quo}}}/64 \otimes \boldsymbol{B}||^2 + \lambda \cdot r(\boldsymbol{u}). \tag{4.5}$$

Note that every bold symbol here, e.g., $\boldsymbol{u}$, represents a 4×4 matrix. For entropy coding, the 4×4 matrix of $\boldsymbol{u}$ will be zig-zag ordered into a 1×16 sequence. To facilitate our following discussion of algorithm design based on CAVLC, we introduce a new denotation, i.e., to add a bar on the top of a bold symbol to indicate the zig-zag ordered sequence of the corresponding matrix. E.g., $\bar{\boldsymbol{u}}$ represents the 1×16 vector obtained by ordering $\boldsymbol{u}$. Then, the equation of (4.5) is rewritten as follows,

$$\bar{\boldsymbol{u}} = \arg\min_{\bar{\boldsymbol{u}}} ||\bar{\boldsymbol{c}} - \bar{\boldsymbol{u}} \otimes \bar{\boldsymbol{dq}}[p_{\text{rem}}] \cdot 2^{p_{\text{quo}}}/64 \otimes \bar{\boldsymbol{B}}||^2 + \lambda \cdot r(\bar{\boldsymbol{u}}),$$

where we still use the symbol $\otimes$ to indicate the element-wise multiplication between two vectors. Finally, for simplicity, we denote $\bar{\boldsymbol{b}}(p) = \bar{\boldsymbol{dq}}[p_{\text{rem}}] \cdot 2^{p_{\text{quo}}}/64 \otimes \bar{\boldsymbol{B}}$ and obtain the following SDQ problem:

$$\bar{\boldsymbol{u}} = \arg\min_{\bar{\boldsymbol{u}}} ||\bar{\boldsymbol{c}} - \bar{\boldsymbol{u}} \otimes \bar{\boldsymbol{b}}(p)||^2 + \lambda \cdot r_{\text{CAVLC}}(\bar{\boldsymbol{u}}). \tag{4.6}$$

Note that the rate function $r(\cdot)$ is further clarified to be related to CAVLC[2].

4.2.2 Graph Design for SDQ based on CAVLC

The minimization problem in (4.6) is equivalent to a search for an output sequence to minimize the rate distortion cost. Targeting an efficient search, we propose a graph-based method. Specifically, we will use the graph shown in Figure 4.1 to represent the vector space of the quantization outputs, with each transition standing for a $(run, level)$ pair and each path from the initial state (denoted as HOS) to the

[2]The clarification is for avoiding confusion with latter discussions about SDQ design based on CABAC.

end state (denoted as EOS) in the graph giving a unique sequence of quantization outputs. As discussed in the above, the distortion term in (4.6) can be easily computed in an element-wise additive manner. However, it is difficult to evaluate the rate term due to the adaptive coding table selection in CAVLC. Graph 4.1 will facilitate an additive rate evaluation according to the CAVLC coding process reviewed in the above. It has 16 columns corresponding to 16 quantization coefficients in addition to the initial and end states with each column further containing a set of states. In the following, we will first define those states and then describe how they are connected to form the graph 4.1.

Definition of states according to CAVLC level coding

CAVLC encodes *level*s based on adaptive contexts, which are used to select VLC tables. These adaptive contexts are represented by different states in Graph 4.1. Let us first examine the trailing one coding rule (see [3] for details). The trailing ones are a set of *level*s with three features. First, they must be handled at the beginning of the coding process. (Note that coding is conducted in the reverse order of the zig-zag sequence.) Second, they are consecutive. Third, there is a restriction of to consider at most 3 of them. To meet these three requirements, we design three types of states, Tni, $i = 1, 2, 3$. In addition, CAVLC requires to know the number of trailing ones, i.e., *T1s*, both at the beginning of the coding process (*T1s* is transmitted) and at the point that the *level* coding table is initialized. As such, we define 6 states, Tn3H, Tn2H, Tn1H, Tn2T, Tn1T, and Tn1TH as shown in Figure 4.2, where TnjH in the column of c_i represents that c_i is the first trailing one and $T1s = j$, TnjT in the column of c_i represents that c_i is the $(4-j)$th trailing one and $T1s = 3$, and Tn1TH in the column of c_i represents that c_i is the second trailing one and $T1s = 2$. Hereafter, these states are also referred to as T-states.

More states are defined based on features for coding *level*s other than trailing ones using CAVLC. The two important factors for coding these *level*s are rate functions for corresponding tables and table selection criteria. For the purpose of

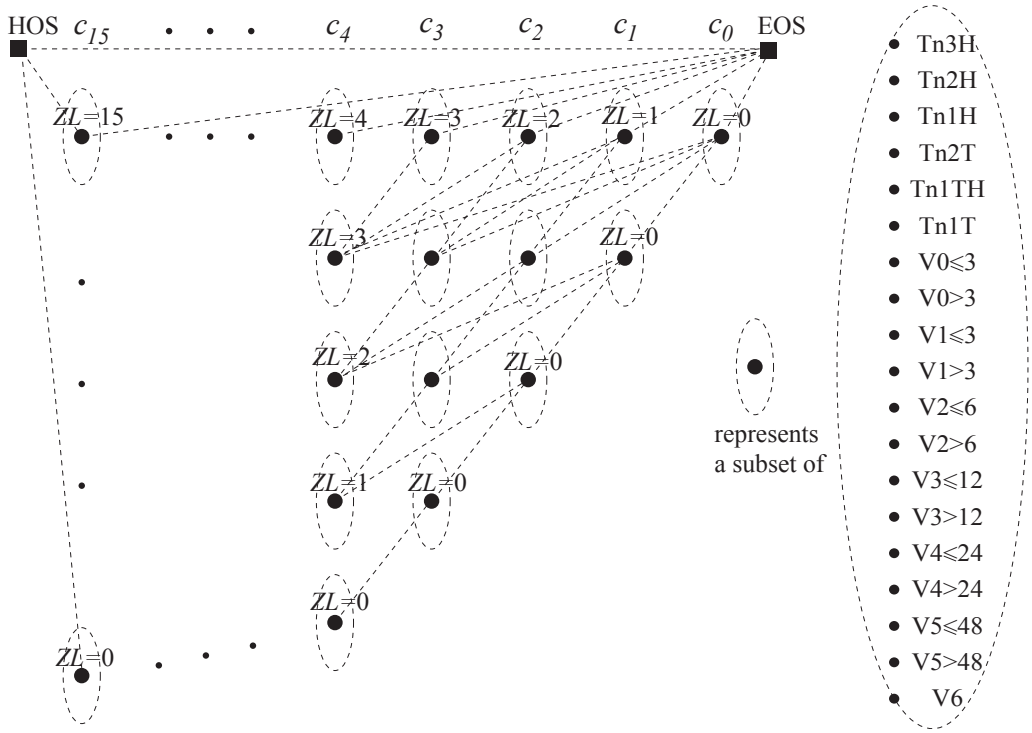


Figure 4.1: The graph structure for SDQ based on CAVLC. There are 16 columns according to 16 coefficients in a 4×4 block. A column consists of multiple state groups, according to different ZL. The left panel shows the connections between these groups. Each group initially contains a set of states defined on the right panel, while eventually only states that receive valid connections remain valid.

rate evaluation, two tables are different only if they have different rate functions. The following equations summarize the rate functions for Vlc(0)-Vlc(6),

$$r(\text{Vlc}(0), u) = \begin{cases} 2u - 1, & 0 < u < 8 \\ -2u, & -8 < u < 0 \\ 19, & 8 \leq |u| \leq 15 \\ 28, & \text{o.w.} \end{cases} \tag{4.7}$$

$$r(\text{Vlc}(i), u) = \begin{cases} \frac{|u|-1}{2^{i-1}} + i + 1, & |u| \leq 15 \cdot 2^{i-1} \\ 28, & \text{o.w.} \end{cases} \quad i = 1, \cdots, 6. \tag{4.8}$$

Now consider the table selection. It is based on a set of thresholds assigned to those codes:

$$T_i = 3 \cdot 2^{i-1}, \quad i = 1, \cdots, 5.$$

Note that the threshold for Vlc(0) is 0, meaning that it always switches to another table. There is no threshold defined for Vlc(6). Once it is selected, it will be used until the end of the current block. Other than these, the coding table will be switched from Vlc(i) to Vlc($i+1$) when the current *level* is greater than T_i. Vlc(0) will switch to Vlc(2) when $level > 3$. Therefore, each coding table except Vlc(6) needs to have two states in order to clearly determine the context to choose a coding table for the next *level* according to the current *level*. As shown in Figure 4.3, there are 13 states defined, named as either V$i \leq T_i$ or V$i > T_i$. These states are refereed to as V-states. The above state definition also implies a restriction to the state output. For example, the output for the state V$i > T_i$ must be greater than T_i. Consider the dynamic range of $[1, 2^{12}]$ for a *level* in H.264. The output range for V$i \leq T_i$ is $[1, T_i]$, while the output for V$i > T_i$ will be any integer in $[T_i + 1, 2^{12}]$. For V6, the output range will be the full range of $[1, 2^{12}]$.

Definition of state groups according to run coding

Now we examine the *run*s coding process of CAVLC and explain why and how states are clustered into groups. The context for choosing a table to code *run*s

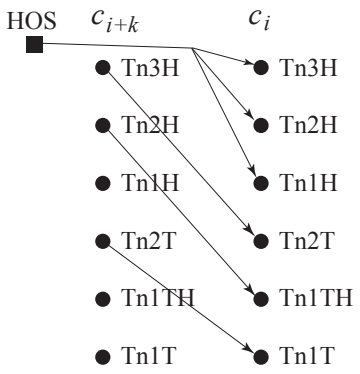


Figure 4.2: States and connections defined according to the trailing one coding rule of CAVLC. HOS is a dummy state, indicating the start of encoding.

depends on the parameter of *ZL*, which is involved in future states in the graph structure. To build this dependency into the definition of states, we define a state group for each different *ZL*. As shown in Figure 4.1, a state group initially consists of all T-states and V-states. For the column of coefficient c_i, there are $(i + 1)$ groups, corresponding to $ZL{=}0, 1, \cdots, i$.

Besides helping the *run* coding table selection, the formation of state groups according to *ZL* provides other two advantages. First, it naturally leads us to know *TotalZeros* for every path in the graph. Second, it enables us to include the coding rate of *CoeffToken* in the optimization process by providing the value of *TC*. In addition, *TC* is also used to initialize the *level* coding table.

Connecting states to build up a graph

Connections from one column to another are now established in two steps. The first is to connect state groups, and the second is to further clarify connections between states in two connected groups. Specifically, HOS is connected to all groups, while a group in the column of c_i is connected to EOS only if its *ZL* equals to i. Moreover, consider the mth group in the column of c_i $(0 \leq m \leq i)$ with *ZL* being m and the nth group in the column of c_j with *ZL* being n, where $i > j$. These two groups are connected if and only if $i - m{=}j - n$. This rule is illustrated in Figure 4.1.

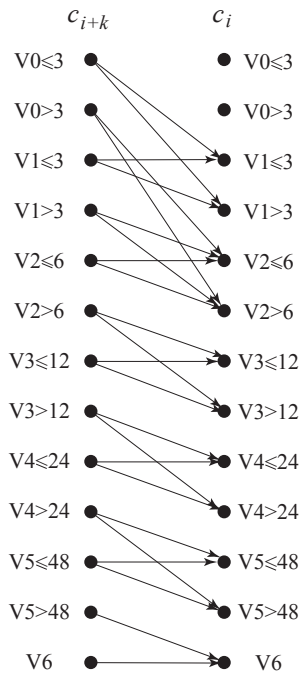


Figure 4.3: States and connections defined according to the *level* coding process of CAVLC.

Now we discuss connections between two groups. First, two rules are defined as Tn3H→Tn2T→Tn1T and Tn2H→Tn1TH between T-states as shown in Figure 4.2. Second, connections between V-states are established by two rules, as illustrated in Figure 4.3:

1. The state V$i \leq T_i$ will go to both V$i \leq T_i$ and V$i > T_i$.
2. The state V$i > T_i$ will go to both V$i+1 \leq T_{i+1}$ and V$i+1 > T_{i+1}$.

Third, we utilize the *level* coding table initialization rule to set up other necessary connections including those from the initial state HOS and those to the end state EOS.

1. Connections from HOS to T-states. HOS is connected to Tn3H in the column corresponding to c_i when $i \geq 2$; HOS is connected to Tn2H in the column corresponding to c_i when $i \geq 1$; HOS is connected to all Tn1H states.

2. Connections from HOS to V-states in a group with ZL in the column corresponding to c_i. This is for the case where $T1s = 0$. Connect HOS to V0 ≤ 3 and V0 > 3 if $i + 1 - ZL \leq 10$. Connect HOS to V1 ≤ 3 and V1 > 3 if $i + 1 - ZL > 10$.

3. Connections from Tn1H to V-states in a group with ZL in the column corresponding to c_i. This is for the case where $T1s = 1$. Connect Tn1H to V0 ≤ 3 and V0 > 3 if $i + 1 - ZL \leq 9$. Connect Tn1H to V1 ≤ 3 and V1 > 3 if $i + 1 - ZL > 9$.

4. Connections from Tn1TH to V-states in a group with ZL in the column corresponding to c_i. This is for the case where $T1s = 2$. Connect Tn1TH to V0 ≤ 3 and V0 > 3 if $i + 1 - ZL \leq 8$. Connect Tn1TH to V1 ≤ 3 and V1 > 3 if $i + 1 - ZL > 8$.

5. Connecting Tn1T to V0 ≤ 3 and V0 > 3.

Eventually, while each group initially contains 19 states as shown in Figure 4.1, only those states that receive valid connections remain. The graph ends at a dummy state EOS.

Metric assignment

Now we discuss parallel transitions before presenting metric assignment to a transition in the graph. Because each state may accord to multiple quantization outputs, there exist multiple transitions between two connected states. As discussed above, there are two types of states, i.e., T-states and V-states. While a T-state clearly outputs 1, the output of a V-state can be any integer within a given range. Accordingly, there exist multiple transitions for a connection to a V-state. Consider a connection from a state s_1 in the column corresponding to c_i to a state s_2 in the column corresponding to c_j. Denote the output range of s_2 as $[u_{\text{low}}, u_{\text{high}}]$. There will be $(u_{\text{high}} - u_{\text{low}} + 1)$ parallel transitions from s_1 to s_2, with each according to a

unique quantization output. Clearly, the only difference between those transitions is the quantization output. However, the output is well constrained within a range so that the difference will not affect any other connections. Therefore, they are named parallel transitions.

Now, we assign metrics to three types of transitions, i.e., a transition starting from HOS, a transition ending at EOS, and a transition from a state s_1 in the column of c_i to another state s_2 in the column of c_j. The metric for a transition from HOS to s_1 in the column of c_i is defined as follows.

$$g_{\text{head}}(c_i, s_1) = \sum_{k=i+1}^{15} c_k^2 + \lambda \cdot r(ZL, T1s, TC) + (c_i - u_i \cdot b_i)^2 + \lambda \cdot r_{s_1}(u_i), \qquad (4.9)$$

where the first term is the distortion for quantizing $c_{15}, \cdots, c_{i+1}$ to zeros as the encoding starts with c_i, the last two terms accord to the RD cost for quantizing c_i to u_i, and b_i is the ith element of the vector $\bar{\boldsymbol{b}}(p)$ as defined in (4.6).

The metric for a transition from s_1 in the column of c_i to s_2 in the column of c_j, $(i > j)$ is defined as

$$g_{\text{n}} = \sum_{k=j+1}^{i-1} c_k^2 + \lambda \cdot r_{s_1}(i-j-1) + (c_j - u_j \cdot b_j)^2 + \lambda \cdot t_{s_2}(u_j), \qquad (4.10)$$

where the first term computes the distortion for quantizing some coefficients to zero, the second term is the rate cost for coding the *run* with $r_{s_1}(i-j-1)$ given by the *run* coding table at state s_1, and the last two terms are the RD cost for quantizing c_j to u_j with $t_{s_2}(u_j)$ determined by the *level* coding table at state s_2.

Finally, for a transition from a state in the column corresponding to c_j to EOS, the RD cost is

$$g_{\text{end}}(c_j) = \sum_{k=0}^{j-1} c_k^2, \qquad (4.11)$$

which accords to the distortion for quantizing all remaining coefficients from c_{j-1} to c_0 to zeros.

4.2.3 Algorithm, Optimality, and Complexity

With the above metric assignments, the problem of (4.6) can be solved by running dynamic programming over Graph 4.1. In other words, the optimal path resulting from dynamic programming applied to Graph 4.1 will give rise to the optimal solution to (4.6), as shown in the following theorem.

Theorem: Given a 4×4 residual block, applying dynamic programming for a search in the proposed graph gives the optimal solution to the SDQ problem of (4.6).

The proof of the above theorem is sketched as follows. For a given input sequence $\bar{\boldsymbol{c}} = (c_{15}, \cdots, c_0)$, any possible sequence of quantization outputs accords to a path in the proposed graph, and vice versa. Define a metric for each transition in the graph as by Equations from (4.9) to (4.11). Carefully examining details of CAVLC will show that the accumulated metric along any path leads to the same value as evaluating the RD cost in (4.6) by really going through CAVLC to code the corresponding output sequence. Thus, when dynamic programming, e.g., the Viterbi algorithm [81], is applied to find the path with the minimize RD cost, the obtained path gives the quantization output sequence to solve (4.6).

The complexity of the proposed graph-based SDQ algorithm (i.e., dynamic programming applied to Graph 4.1) mainly depends on three factors, i.e., the number of columns as 16, the number of states in each column, and the number of parallel transitions for each connection. Expansion of Graph 4.1 into a full graph reveals that the number of states at various columns varies from 17 to 171. With states selectively connected, the major computational cost is to handle the parallel transitions. For a connection from a state s_1 in one column to a state s_2 in another column, the number of parallel transitions is $(u_{\text{high}} - u_{\text{low}} + 1)$, where $[u_{\text{low}}, u_{\text{high}}]$ is the range of all possible quantization outputs at the state s_2. From (4.9) and (4.10), it follows that the only difference among the RD costs assigned to these parallel transitions is in the RD costs arising from different quantization outputs $u \in [u_{\text{low}}, u_{\text{high}}]$. Studies on CAVLC show the rate variation due to different $u \in [u_{\text{low}}, u_{\text{high}}]$ is in-

Table 4.1: Statistics corresponding to 6 parallel transitions in H.264 baseline profile optimization.

	floor(u)-2	floor(u)-1	floor(u)	ceil(u)	ceil(u) +1	ceil(u) +2
Occurrences	0	13644	154328	110268	16955	0

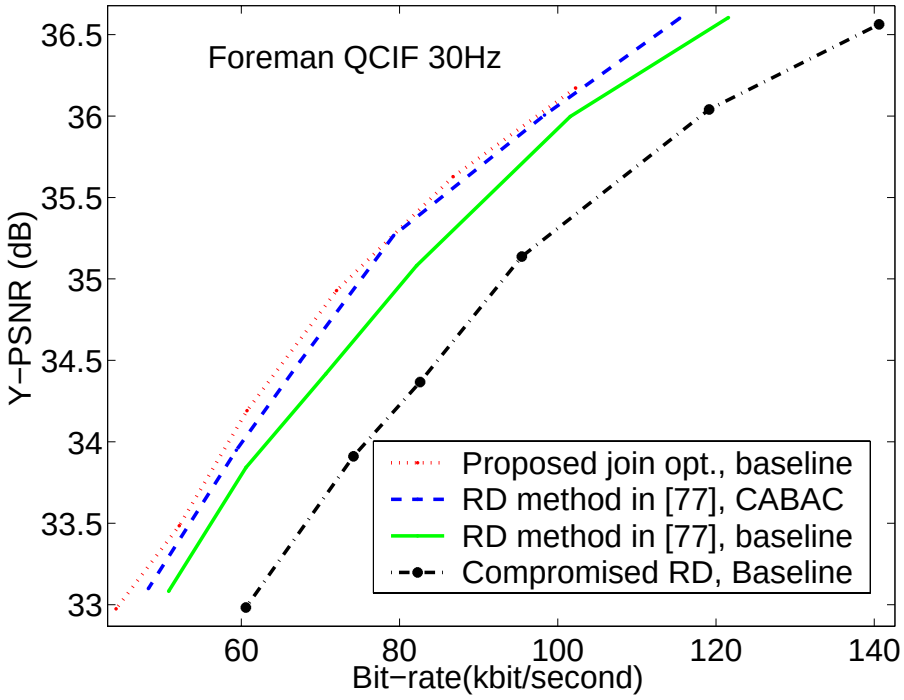
significant compared to the quadratic distortion. This implies that the quantization output for the optimal transition is very likely within a small neighboring region around the hard-decision quantization output $\hat{u} \in [u_{\text{low}}, u_{\text{high}}]$ that minimizes the quadratic distortion. Thus the number of parallel transitions to be examined in practice could be much smaller. Table 4.1 shows the result of an experiment, in which we collect the number of occurrences for events that a real-valued coefficient u is quantized to 6 integers around it. It is shown that it is sufficient to compare 4 parallel transitions around $\hat{u}$, and hence the complexity is reduced to a fairly low level.

4.3 Experimental Results

Experiments have been conducted to study the coding performance of the proposed RD method for optimizing H.264 baseline profile coding. The algorithms are implemented based on the H.264 reference software Jm82[42]. Each sequence is divided into and encoded by groups of frames. In each group, there is one standard I-frame, while all the subsequent frames are coded as P-frames. Experiment results are reported with a group size of 21. The range for full-pixel motion estimation is ± 32, and 5 reference frames are used for motion estimation.

Comparative studies of the coding performance are shown by RD curves, with the distortion being measured by $PSNR$ defined as $PSNR = 10 \log_{10}(255^2)/MSE$, where MSE is the mean square error. Given two methods A and B, a so-called relative rate saving of B to A is computed as in [77] by,

$$S(PSNR) = 100 \cdot \frac{R_{\text{A}}(PSNR) - R_{\text{B}}(PSNR)}{R_{\text{A}}(PSNR)}\%,$$



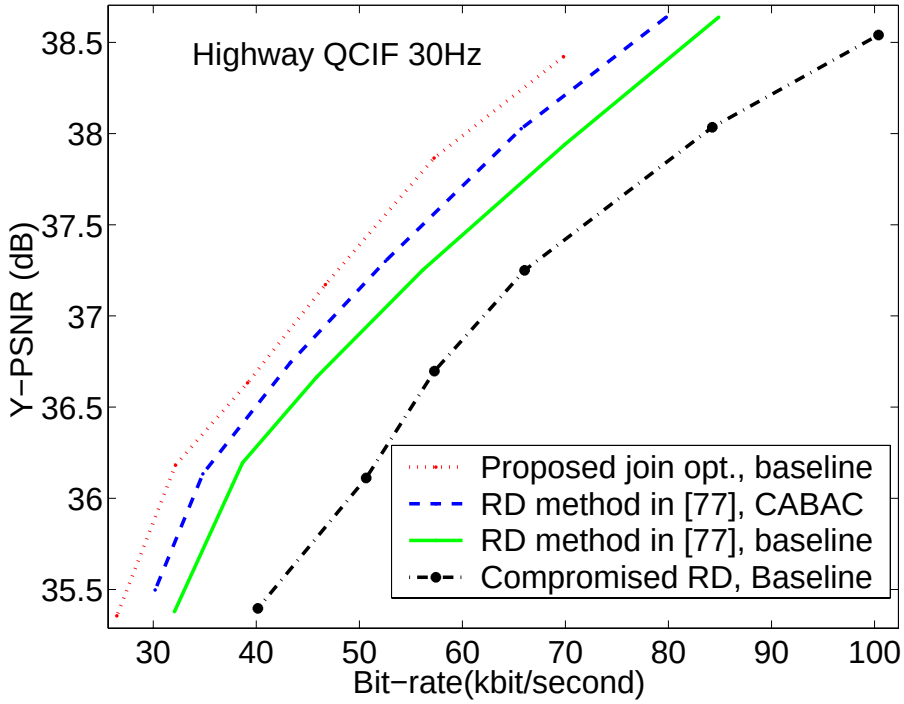


Figure 4.4: The RD curves of four coding methods for coding video sequences of "Foreman" and "Highway".

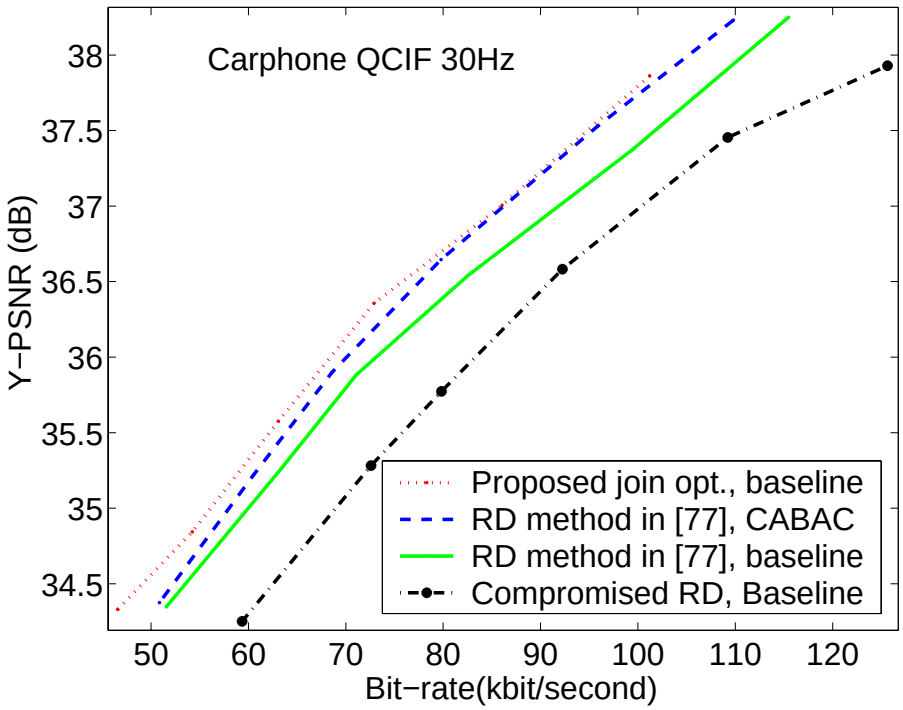


Figure 4.5: The RD curves of four coding methods for coding the video sequence of "Carphone".

where $R_{\mathrm{A}}(PSNR)$ and $R_{\mathrm{B}}(PSNR)$ are the rate corresponding to a given $PSNR$ for methods A and B, respectively. $R_{\mathrm{A}}(PSNR)$ and $R_{\mathrm{B}}(PSNR)$ are calculated by interpolations based on corresponding RD curves. Figures 4.4 and 4.5 show the RD curves for coding various sequences. The RD performance is measured over P-frames only since I-frames are not optimized. The result is reported on the luma component as usual. Comparisons are conducted among four encoders, i.e., a baseline encoder with the proposed overall joint optimization method, a main-profile reference encoder with the RD optimization method in [77] and CABAC (the coding setting of this encoder is the same as that of a baseline profile except that CABAC is used instead of CAVLC), a baseline reference encoder with the RD optimization method in [77], and a baseline reference encoder with compromised RD optimization[3]. The RD curve for the proposed method is obtained by

[3]This is conducted by disabling the RD optimization option in the JM software. In this case, empirical formulas are used to compute the RD cost for mode selection, resulting in a compromised

varying the slope λ, while RD curves for other methods result from varying the quantization step size. Specifically, the six points on the curve of the proposed joint optimization method accord to $\lambda = \{27.2, 34.3, 43.2, 54.4, 68.5, 86.3\}$. As illustrated in Figures 4.4 and 4.5, the baseline encoder with the proposed overall joint optimization method achieves a significant rate reduction over the baseline reference encoder with the RD optimization in [77]. Moreover, experiments over a set of 8 video sequences (i.e., Highway, Carphone, Foreman, Salesman, Silent, Container, Mother-Daughter, Grandma) show the proposed joint optimization method achieves an average 12% rate reduction while preserving the same PSNR over the RD optimization in [77] with the baseline profile, and 23% rate reduction over the baseline encoder with compromised RD optimization.

It is interesting to compare the proposed joint optimization method using CAVLC and the method in [77] using CABAC. Theoretically, CABAC holds advantage over CAVLC by its adaptability to the symbol statistics and its ability to use a noninteger code length. The fundamental 1bit/symbol limit on variable length codes leads to a poor coding performance for CAVLC when the symbol probability is large. Surprisingly, this fundamental deficit of CAVLC to CABAC has been well compensated when we tune up the whole system with the joint optimization. Indeed, as shown in Figures 4.4 and 4.5, the joint optimization method using CAVLC slightly outperforms the method in [77] using CABAC. Since CAVLC is faster than CABAC for decoding, the proposed joint optimization method results in a codec with a better coding performance and faster decoding while compared to the method in [77] using CABAC.

Figure 4.6 compares the coding gain among the proposed three algorithms. For simplicity, the encoders with three proposed algorithms are referred to as Enc(SDQ), Enc(SDQ+QP), and Enc(SDQ+QP+ME), respectively, while the forth encoder is called Enc(baseline, [77]). For Enc(SDQ), motion estimation and quantization step sizes are computed using the baseline method in [77]. For Enc(SDQ+QP),

RD performance.

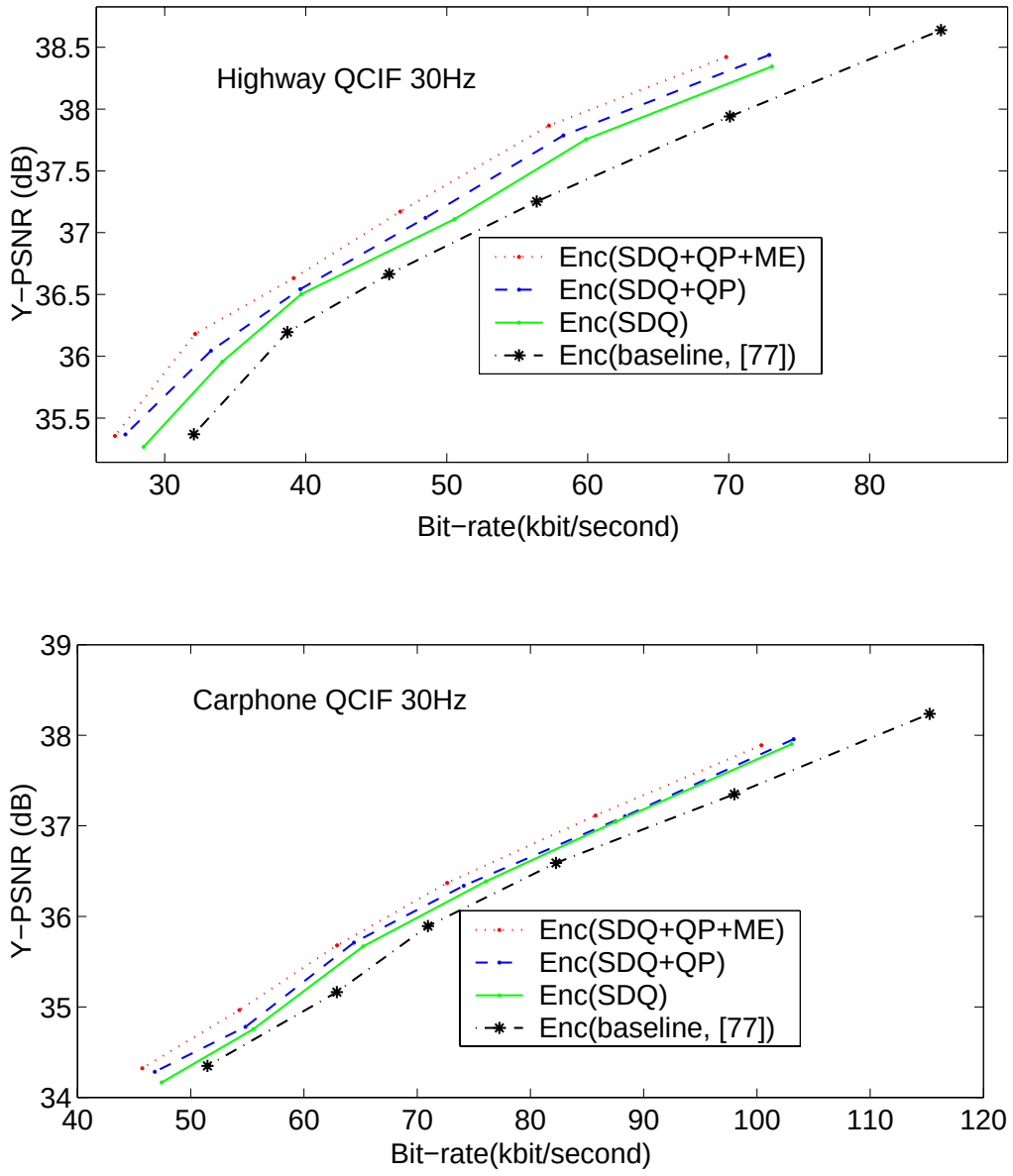


Figure 4.6: Comparison of the coding gain for the proposed three algorithms, Enc(SDQ), Enc(SDQ+QP), and Enc(SDQ+QP+ME), with H.264 baseline profile compatibility.

the proposed residual coding optimization is performed based on the motion estimation obtained using the baseline method in [77]. It is shown that approximately, half of the gain for overall joint optimization comes from SDQ [4], while QP and ME contribute the other half gain together. On average, our experiments show rate reductions of 6%, 8%, and 12% while preserving PSNR by Enc(SDQ), Enc(SDQ+QP), and Enc(SDQ+QP+ME), respectively, over Enc(baseline, [77]).

In term of program execution time with our current implementation, the baseline encoder using RD optimization of [77] takes 1 second to encode a P frame. SDQ adds 1 second for each P frame. QP+SDQ takes 6 seconds to encode each frame. The overall optimization with SDQ+QP+ME takes 15 seconds per frame. The complexity of SDQ+QP comes from the process to explore a neighboring region of 5 quantization step sizes. The complexity of the overall algorithm mainly comes from the iterative procedure, for which two iterations are used. Frankly, the current implementation is not efficient and there is plenty of room to improve the software structure and efficiency. Meanwhile, compared with the RD method in [77] and the compromised RD method, the proposed approach seeks for better RD performance while maintaining the decoding complexity. It targets off-line applications such as video delivery, for which the RD performance is more important and a complicated encoder is normally acceptable since encoding is carried out only once.

The proposed joint optimization algorithm works in a frame-by-frame manner. Clearly, the optimization of the current P-frame encoding will impact on the coding of the next P-frame. Thus, it is interesting to see such impact as the number of optimized P-frames increases. Figure 4.7 shows the results of the relative rate savings of the proposed joint optimization algorithm over the baseline reference encoder with compromised RD optimization for various numbers of P-frames. Also shown in Figure 4.7 is the result for the RD method in [77]. Although the proposed

[4]It may be interesting to relate the SDQ gain to the picture texture. In general, they can be related to each other qualitatively through the effectiveness of motion estimation. I.e., the gain from SDQ is higher when the energy of residual signals is greater. Usually, this accords to a less effective motion estimation, which may be observed for highly textured pictures.

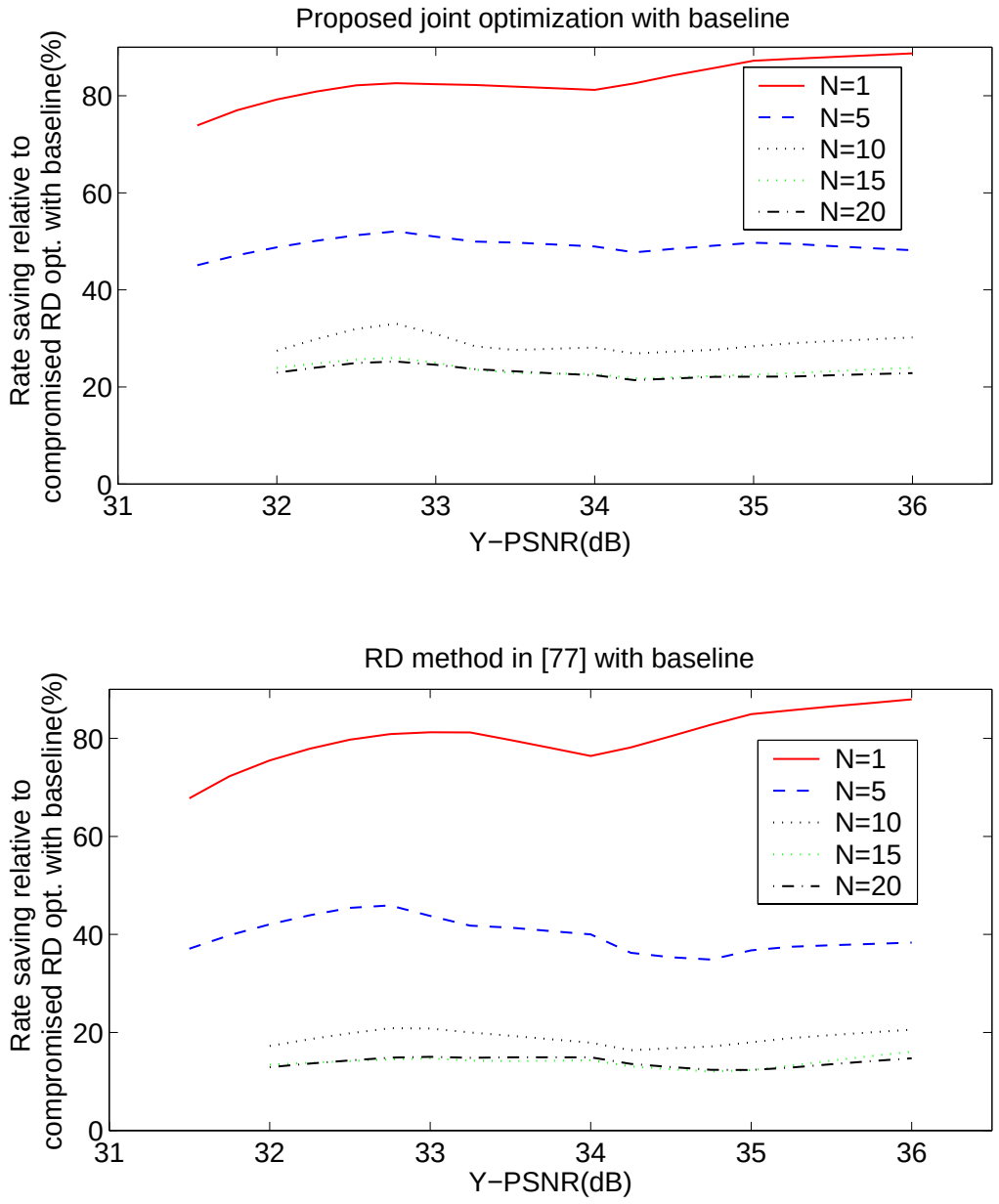


Figure 4.7: The relative rate savings averaged over various numbers of frames for coding the sequence of "Salesman".

joint optimization algorithm constantly provides better gains than the RD method in [77], the relative rate savings decreases as N increases in both cases. This warrants the joint optimization of a group of frames, which is left open for future research.

4.4 Chapter Summary

In this chapter, we have applied the framework proposed in Chapter 3 to optimize RD trade-off for H.264 baseline profile encoding. Particularly, a graph-based SDQ algorithm has been developed based on CAVLC. It has been shown that if the weak adjacent block dependency utilized in CAVLC of H.264 is ignored for optimization, the proposed graph-based SDQ algorithm is indeed optimal and so is the algorithm for residual coding. These algorithms have been implemented based on the reference encoder JM82 of H.264 with complete compatibility to the baseline profile. Experiments have demonstrated that for a set of typical video testing sequences, the graph-based SDQ algorithm, the algorithm for residue coding, and the iterative overall algorithm achieve on average, 6%, 8%, and 12%, respectively, rate reduction at the same PSNR (ranging from 30dB to 38dB) when compared with the RD optimization method implemented in the H.264 baseline reference software.

As discussed in Chapter 3, the proposed optimization framework is applicable to any hybrid video coding scheme. In the following, we study its application to optimizing H.264 main profile encoding.

Chapter 5

RD Optimal Coding with H.264 Main Profile Compatibility

This chapter is focused on applying our proposed RD optimization framework to optimizing RD trade-off for H.264 main profile encoding. Specifically, an SDQ algorithm is proposed based on the entropy coding method CABAC and experiments are conducted to verify the performance. In the following, CABAC is reviewed before a graph structure is designed based on it for SDQ.

5.1 Review of CABAC

CABAC consists of three steps [51]:

1. Binarization. The so-called UEG0 algorithm is used to convert non-zero transform coefficient levels into a binary representation so that the binary arithmetic coding engine can be used to code them.

2. Context modeling. CABAC defines a probability model for each binary bit. In the following, we will discuss those related to our SDQ design.

3. Binary arithmetic coding. The binary representation is encoded bit by bit using corresponding models.

The SDQ design to be presented is closely related to the context modeling for residual coding. Residual coding by CABAC includes two parts, i.e., coding of a so-called significance map and coding of non-zero coefficients. Given a zig-zag ordered sequence of transform coefficient levels, its significance map contains a binary sequence of significant coefficient flags and a sequence of last significant coefficient flags. The context modeling for coding the significance map is associated with the zig-zag order and is easy to be included in the soft decision design. The context modeling for coding non-zero coefficients, however, is complicated. For a given sequence, there are in total 10 contexts for coding the absolute values of non-zero coefficients, with 5 of them for coding the first bit of a binary representation and the other 5 dedicated to coding bits from the second to the 14th. Briefly, these contexts are selected as follows,

1. For a given nonzero transform coefficient level, check the coded part of the sequence. Compute *NumLg1* as the number of coded levels that are greater than 1 and *NumEq1* as the number of coded levels that equal to 1.

2. Determine the context for coding the first bit, named pin1, of the binary representation for the current level as,

$$Ctx_{\text{pin1}} = \begin{cases} 0, & NumLg1 > 0 \\ \min(4, 1+NumEq1), & \text{otherwise.} \end{cases}$$

3. The context for the 2nd $\sim$ 14th bits is selected by

$$Ctx_{\text{pin2}} = \min(4, NumLg1).$$

There is also a bypass mode with a fixed distribution. The remaining bits after the 15th, as well as the sign bits, are coded using the bypass mode. Table 5.1 shows a simple example of the CABAC encoding. Note that the encoding is carried out in the reverse order of the zig-zag scan.

Table 5.1: A simple example for CABAC significance map encoding and context modeling.

Scanning position	1	2	3	4	5	6	7	8	9
Transform coefficient levels	-8	-4	1	0	0	0	0	0	-1
Significant coefficient flag	1	1	1	0	0	0	0	0	1
Last coefficient flag	0	0	0						1
NumLg1	1	0	0						0
NumEq1	2	2	1						0
Ctx_{pin1}	0	3	2						1
Ctx_{pin2}	1	0							

5.2 SDQ Design based on CABAC

In this section, we develop an SDQ algorithm based on CABAC in the main profile of H.264.

5.2.1 Graph Design for SDQ based on CABAC

First look at the computation issue associated with the distortion term in (3.12) as it contains the inverse DCT transform. This is the same issue as we have discussed in Section 4.2.1 while designing SDQ based on CAVLC. Actually, the block-dependency decoupling discussion and the resulting formula of (4.1) in Section 4.2 are also valid here for tackling the SDQ design based on CABAC. Meanwhile, the result of distortion computation in (4.6) discussed in Section 4.2.1 is also applicable here because in both cases the transform and quantization parts are the same. Essentially, our SDQ design based on CABAC starts with a formula as follows,

$$\bar{\boldsymbol{u}} = \arg\min_{\bar{\boldsymbol{u}}} ||\bar{\boldsymbol{c}} - \bar{\boldsymbol{u}} \otimes \bar{\boldsymbol{b}}(p)||^2 + \lambda \cdot r_{\text{CABAC}}(\bar{\boldsymbol{u}}), \tag{5.1}$$

which is similar with (4.6), except that the rate function $r(\cdot)$ here accords to CABAC while in (4.6) it is related to CAVLC.

Compared with SDQ design based on CAVLC in Chapter 4, SDQ design based on CABAC is more complicated because CABAC employs an adaptive context updating scheme besides the adaptive context selection scheme, i.e., context models are updated after each symbol has been encoded using CABAC. Thus, the context states, i.e., probabilities in a context model for coding a given level, are dependent on all previously encoded levels. To tackle this problem, we consider to decompose the problem (5.1) into a two-step optimization as follows,

$$\min_{\bar{\boldsymbol{u}}} \min_{\Omega} \quad ||\bar{\boldsymbol{c}} - \boldsymbol{u} \otimes \bar{\boldsymbol{b}}(p)||^2 + \lambda \cdot r(\bar{\boldsymbol{u}}|\Omega), \tag{5.2}$$

where Ω represents context states, or the probabilities in all context models used for coding non-zero transform coefficient levels $\bar{\boldsymbol{u}}$. This decomposition enables an iterative solution to (5.1), in which the objective function is optimized over $\bar{\boldsymbol{u}}$ and Ω alternately. Specifically, the iteration goes as follows,

1. Fix the context states Ω and optimize the RD cost over the quantization outputs $\boldsymbol{u}$, i.e.,

$$\bar{\boldsymbol{u}} = \arg\min_{\bar{\boldsymbol{u}}} \ ||\bar{\boldsymbol{c}} - \boldsymbol{u} \otimes \bar{\boldsymbol{b}}(p)||^2 + \lambda \cdot r(\bar{\boldsymbol{u}}|\Omega), \text{ with given } \Omega \tag{5.3}$$

2. Update context states Ω by the obtained quantization outputs $\bar{\boldsymbol{u}}$.

Clearly, the second step is simple. The main challenge now turns to solve (5.3), for which a graph-based design is proposed in the following.

A Graph Design based on CABAC Encoding

To solve the problem (5.3), we develop a graph structure, in which the rate function $r(\bar{\boldsymbol{u}}|\Omega)$ with given Ω is computed additively.

As shown in Figure 5.1, a graph is constructed based on coding features of CABAC. Basically, states are defined based on the context model selection, which

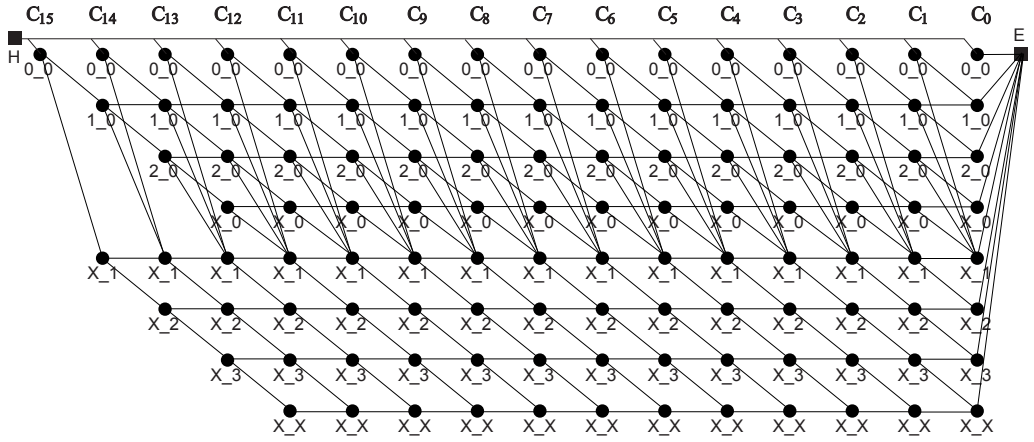


Figure 5.1: The graph structure for SDQ based on CABAC in the main profile of H.264.

depends on two parameters *NumEq1* and *NumLg1*. Thus, states are named by values of *NumEq1* and *NumLg1*, in the form of *NumEq1_NumLg1*, e.g., 2_0 accords to *NumEq1* = 2 and *NumLg1* = 0. When *NumLg1* > 0, the context is irrelevant with *NumEq1*. Thus, there are three states as X_1, X_2, and X_3. The context is fixed for all *NumLg1* ≥ 4. Accordingly, one state X_X is defined. For a 4×4 luma block, there are 16 columns with each of them corresponding to one coefficient. In each column there are up to 8 states. Transitions are established between states according to the increase of *NumEq1* and *NumLg1*, e.g., the state 1_0 is connected to 1_0, 2_0, or X_1 according to quantization outputs of 0, 1, or greater than 1, respectively. In case that quantization outputs are greater than 1, parallel transitions are established so that each accords to a unique value. In practice, because the distortion is a quadratic function with respect to the quantization output, it is sufficient to investigate only a few parallel transitions. Thus the complexity is greatly reduced without sacrificing the RD performance. Finally, a graph structure as shown in Figure 5.1 is obtained.

Rate Distortion Metric Computation in the Graph

Consider a transition from the state H to the mth state at the coefficient c_i and denote it as $s_{i,m}$. Note that $s_{i,m}$ cannot output 0 because any transition from H must go to a so-called last significant coefficient. Denote $r_{\mathrm{s}}(\cdot|\Omega)$, $r_{\mathrm{l}}(\cdot|\Omega)$, and $r_{\mathrm{c}}(\cdot|\Omega)$ as the coding rate for a significant-coefficient-flag bit, a last-coefficient-flag bit, and a quantized coefficient u_i, respectively. Define a metric for this transition as follows,

$$g_{m,i} = (c_i - b_i \cdot u_i)^2 + \lambda \cdot (r_{\mathrm{s}}(1|\Omega) + r_{\mathrm{l}}(1|\Omega) + r_{\mathrm{c}}(u_i|\Omega)). \tag{5.4}$$

Note that both the significant-coefficient-flag bit and the last-coefficient-flag bit are 1.

Further consider a transition from the mth state $s_{i+1,m}$ at coefficient c_{i+1} to the nth state $s_{i,n}$ at coefficient c_i. There are multiple parallel transitions. Different metrics are assigned to transitions with output zero and transitions with outputs greater than zero. Specifically,

$$g_{n,m,i} = \begin{cases} (c_i - b_i \cdot u_i)^2 + \lambda \cdot (r_{\mathrm{s}}(1|\Omega) + r_{\mathrm{l}}(0|\Omega) + r_{\mathrm{c}}(u_i|\Omega)), & u_i \geq 1 \\ c_i^2 + \lambda \cdot (r_{\mathrm{s}}(0|\Omega) + r_{\mathrm{l}}(0|\Omega)), & u_i = 0 \end{cases} \tag{5.5}$$

where the significant-flag bit is 0 or 1 for $u_i = 0$ or $u_i > 0$ and the last-coefficient-flag bit is always 0.

Given selected context models with fixed context states, the rate functions of $r_{\mathrm{s}}(\cdot|\Omega)$, $r_{\mathrm{l}}(\cdot|\Omega)$, and $r_{\mathrm{c}}(\cdot|\Omega)$ in (5.4) and (5.5) are estimated as the self-information of the corresponding probability event. Specifically, context states in CABAC are specified by a pair of (LPS, σ), where LPS indicates the least probable symbol, and $\sigma = 0, \cdots, 63$. Correspondingly, the probability for LPS is specified as[51],

$$p_\sigma(\mathrm{LPS}) = \frac{1}{2} \cdot 0.0375^{\sigma/63}. \tag{5.6}$$

Then, for a selected context model with $(\mathrm{LPS}, \sigma) \in \Omega$ and an input bit $\dot{b}$, the rate is estimated by

$$r_{\mathrm{context}(\mathrm{LPS},\sigma)}(\dot{b}) = \begin{cases} -\log_2(p_\sigma(\mathrm{LPS})) & \dot{b} = \mathrm{LPS} \\ -\log_2(1 - p_\sigma(\mathrm{LPS})) & \dot{b} \neq \mathrm{LPS} \end{cases} \tag{5.7}$$

This estimation is applicable to $r_{\mathrm{s}}(\cdot|\Omega)$, $r_{\mathrm{l}}(\cdot|\Omega)$, and $r_{\mathrm{c}}(\cdot|\Omega)$ all in the same way, except that different context models are selected.

5.2.2 Algorithm, Optimality, and Complexity

Based on the graph design and the metric computation discussed above, the solution to (5.3) now becomes a problem of searching for a path in the graph for the minimal RD cost. It is not hard to see that the proposed graph design would allow an element-wise additive computation of the RD cost in (5.3) with given Ω. In this case, the Viterbi algorithm can be used to do the search. Overall, the SDQ algorithm for solving (5.1) is summarized as follows,

1. Initialize all context states at each column in Figure 5.1 by extracting context states for the current block, and updating it according to the HDQ outputs.

2. Fix context states at each column, and search for a path with the minimal RD cost using Viterbi algorithm.

3. Update context states at each column using the quantization outputs corresponding to the path obtained in Step 2. Repeat Step 2 until the algorithm converges, meaning that the resulted path does not change.

Observations show that the above algorithm converges mostly by two iterations. Table 5.2 shows a simple example of SDQ based on CABAC. The real quantization outputs correspond to the real value of $\mathrm{T}(\boldsymbol{z})/q$ in (3.7). The offset value $\delta = 1/6$ is as used in the reference codec JM82[1]. Note that the value of 1.12 is quantized to 0 by SDQ, which would never happen if HDQ is used.

In general, the optimality of the above SDQ algorithm for (5.1) is not guaranteed due to its iterative nature. Nevertheless, it can be shown that the proposed graph

[1]Adaptive rounding offset has been proposed for H.264 in JVT-N011[54], where the rounding offsets vary for each *level*. However, the quantization with adaptive rounding offset is still considered as HDQ, which is incapable of generating SDQ outputs, as shown by the quantization of the value of 1.12 in Table 5.2.

Table 5.2: An example of SDQ. See explanation of the first column in the text.

Real quantization outputs	-8.13	-4.53	0.88	0.20	0.65	-0.48	-0.58	0.60	-1.12
HDQ outputs by (3.7), $\delta = \frac{1}{6}$	-8	-4	1	0	0	0	0	0	-1
SDQ outputs by one iteration	-9	-4	1	0	0	0	0	0	0
SDQ outputs by two iterations	-8	-4	1	0	0	0	0	0	0

design leads to the optimal solution to (5.3). Thus the SDQ algorithm is referred to as being near-optimal for solving (5.1). Specifically, we summarize the optimality for solving (5.3) in the following theorem.

Theorem: For a 4×4 residual block $\boldsymbol{z}$, the proposed graph design in Figure 5.1 provides the optimal solution to the RD minimization problem defined in (5.3).

The graph represents the whole vector space of quantization outputs and each path in the graph gives a unique block of quantization outputs. Therefore, to prove the theorem is to find a metric for each transition in the graph so that for any path the accumulated metric equals to the RD cost of (5.3). Consequently, Viterbi algorithm can be used to search for a path in the graph to minimize the RD cost and the obtained path gives the optimal quantization outputs for solving (5.3).

As shown in Figure 5.1, at each state the values of *NumEq1* and *NumLg1* are clearly defined, leaving no ambiguity of context selection for coding the non-zero coefficients. Meanwhile, context models for coding the significance map is also known for each state. Therefore, $r_{\mathrm{c}}(\cdot|\Omega)$, $r_{\mathrm{s}}(\cdot|\Omega)$ and $r_{\mathrm{l}}(\cdot|\Omega)$ can be computed using (5.7). By examining the details of CABAC, it is not hard to see that for any given path and its corresponding coefficient sequence, the accumulated metric along the path by (5.4) and (5.5) equals to the result as calculating the RD cost in (5.3) with given Ω. Thus, applying Viterbi algorithm to search the graph leads to the solution of the problem in (5.3).

The complexity of the proposed graph-based SDQ algorithm (i.e., dynamic programming applied to Graph 5.1) depends on four factors, i.e., the number of columns as 16, the number of states in each column as 8, the number of iterations

Table 5.3: Statistics corresponding to 6 parallel transitions in H.264 main profile optimization.

	floor(u)-2	floor(u)-1	floor(u)	ceil(u)	ceil(u) +1	ceil(u) +2
Occurrences	0	18400	162345	129876	17923	0

as 2, and the number of parallel transitions for each connection. Parallel transitions are defined for states with quantization outputs greater than 1. Specifically, the number of parallel transitions is $(u_{\text{high}} - u_{\text{low}} + 1)$, where $[u_{\text{low}}, u_{\text{high}}]$ is the range of all possible quantization outputs. In practice, because the distortion is a quadratic function with respect to the quantization output, the quantization output for the optimal transition is within a small neighboring region around the hard-decision quantization output $\hat{u} \in [u_{\text{low}}, u_{\text{high}}]$. Thus the number of parallel transitions to be examined in practice is small. Table 5.3 shows the result of an experiment, in which we collect the number of occurrences for events that a real-valued coefficient u is quantized to 6 integers around it. It is shown that it is sufficient to compare 4 parallel transitions around $\hat{u}$, and hence the complexity is reduced to a fairly low level.

5.3 Experimental Results

The proposed joint optimization method is implemented based on the H.264 reference software Jm82. Only the first frame is intra coded (I-frame), while all the subsequent frames use temporal prediction (P-frame). The range for full-pixel motion prediction is ± 32. The iteration for the joint optimization is stopped when the RD cost decrease is less than 1%. Comparative studies of the coding performance are shown by RD curves. with the distortion being measured by PSNR as defined in Chapter 4. The RD performance is measured over P-frames only since I-frames are not optimized. As usual, the result is reported on the luma component.

Figure 5.2 shows the RD performance for coding two typical video sequences

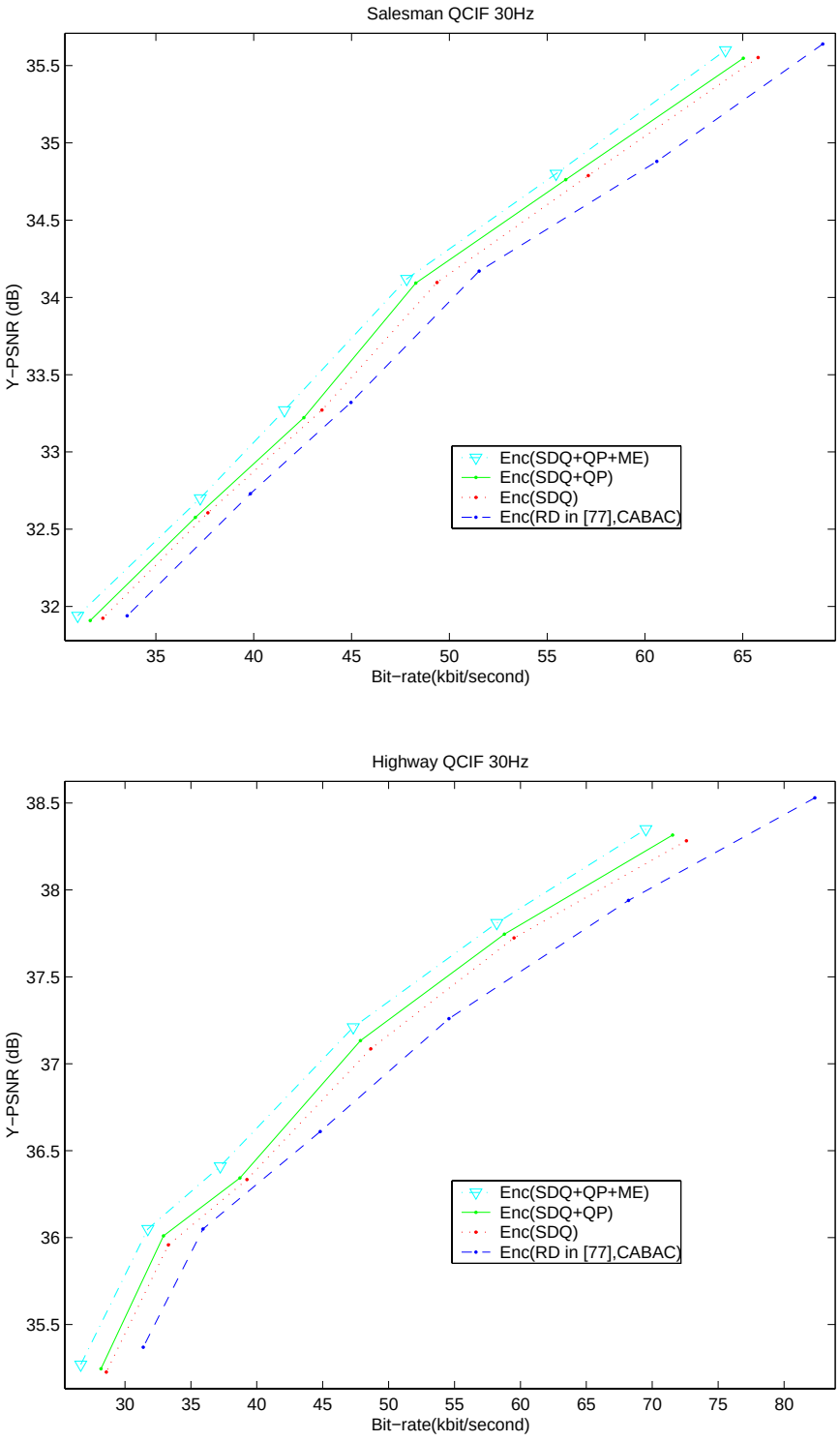


Figure 5.2: RD performance for coding "Salesman.qcif" and "highway.qcif", corresponding to the three algorithms in the proposed RD optimization framework and a main profile reference encoder with the RD optimization method in [77] and CABAC.

using the proposed SDQ algorithm based on CABAC. Furthermore, it also shows the RD performance for embedding the SDQ into the residual coding optimization algorithm and that for embedding the residual coding optimization algorithm into the overall optimization algorithm. For simplicity, the encoders are referred to as Enc(SDQ), Enc(SDQ+QP), and Enc(SDQ+QP+ME), while the encoder with the RD optimization method in [77] and CABAC is called Enc(RD in [77], CABAC). For Enc(SDQ), motion estimation and quantization step sizes are computed using the main profile method in [77]. For Enc(SDQ+QP), the residual coding optimization is performed based on the motion estimation obtained using the main profile method in [77]. It is shown that, approximately, half of the gain by the joint optimization comes from SDQ. Specifically, our experiments show on average rate reductions of 5%, 7%, and 10% while preserving the same PSNR by Enc(SDQ), Enc(SDQ+QP), and Enc(SDQ+QP+ME), respectively, over Enc(RD in [77], CABAC).

In Figures 5.3 and 5.4, we further compare the RD performance of our joint optimization method in this study with other four methods, i.e., a baseline encoder with joint RD optimization implemented in [34], a main-profile reference encoder with the RD optimization method in [77] and CABAC (the coding setting of this encoder is the same as that of a baseline profile except that CABAC is used instead of CAVLC), a baseline reference encoder with the RD optimization method in [77], and a baseline reference encoder with compromised RD optimization. Figures 5.3 and 5.4 show that the joint optimization method in this study achieves the best RD performance among 5 methods mentioned above. Experiments in [34] showed that the joint design based on the baseline profile with CAVLC outperformed the method based on the main profile with CABAC in [77]. In this study, it is shown that the joint design based on the main profile with CABAC results in a better RD performance than the joint design based on the baseline method CAVLC, which is as expected since CABAC is superior to CAVLC. Specifically, it is interesting to see that an average 10% rate reduction is achieved by the joint optimization method based on the proposed SDQ for CABAC over the main-profile reference

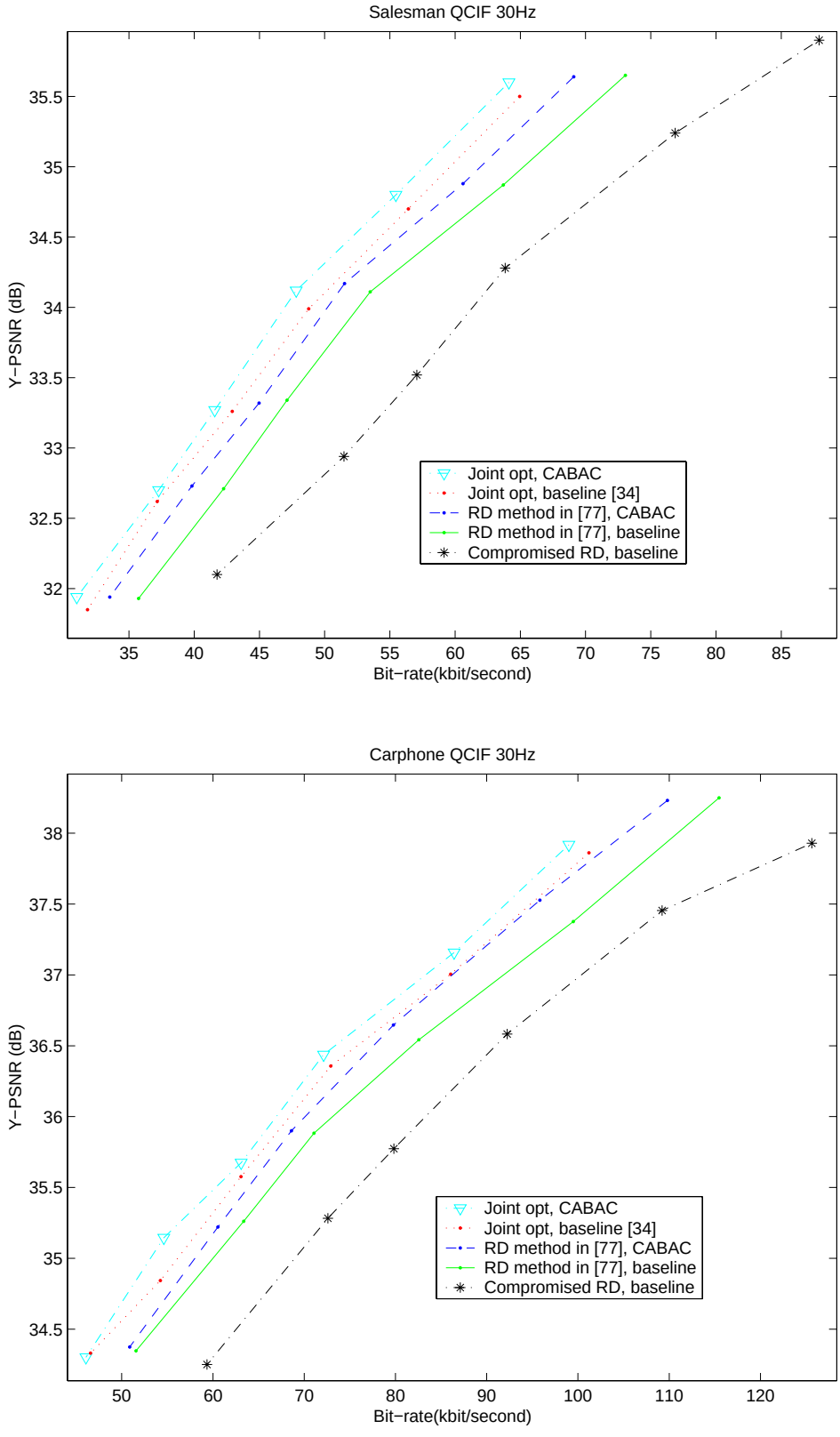


Figure 5.3: The RD curves for coding "Salesman.qcif" and "Carphone.qcif" corresponding to five H.264 main profile compliant encoders.

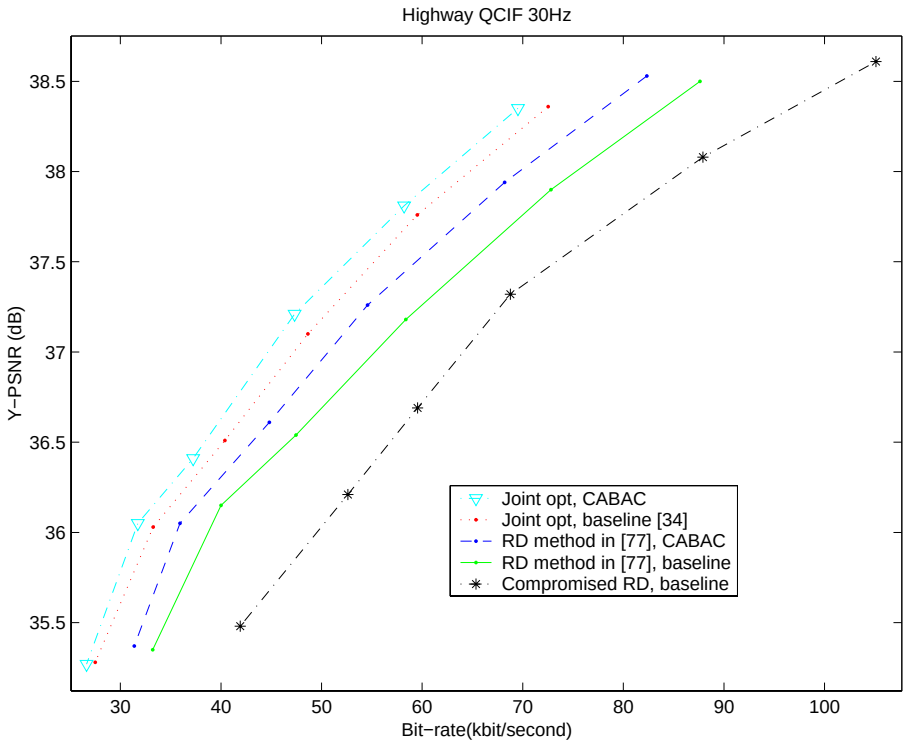


Figure 5.4: The RD curves for coding "highway.qcif" corresponding to five H.264 main profile compliant encoders.

encoder with the RD optimization method in [77] and CABAC. If compared with the baseline reference encoder with RD optimization in [77], the proposed joint optimization method with SDQ based on CABAC yields a 20% rate reduction, while the rate gain[2] by the proposed joint optimization method with SDQ based on CAVLC is 12%, as presented in Chapter 4.

In term of program execution time with our implementation, the complexity of the joint optimization with SDQ based on CABAC is similar with that of the joint optimization with SDQ based on CAVLC. Specifically, a main-profile reference encoder with CABAC and the RD optimization method in [77] takes a little more than 1 second to encode a P frame, while the overall joint optimization based on the proposed SDQ design takes around 15 seconds per frame. Frankly, the current implementation is not efficient, yet similar arguments as those we made in Chapter 4 may also be made here to justify the proposed method with SDQ based on CABAC. Basically, it targets off-line applications such as video delivery, for which the RD performance is more important and a complicated encoder is normally acceptable since encoding is carried out only once. Furthermore, the proposed method here also helps to satisfy a desire for pushing the coding performance of a standard-compatible codec to its theoretic limit or to achieve the best known coding performance.

5.4 Chapter Summary

In this chapter, the joint RD optimization framework proposed in chapter 3 has been applied to improve H.264 encoding with its main profile decoding compatibility. Based on CABAC, a graph-based SDQ design has been developed, which forms the core for jointly optimizing motion prediction, quantization, and entropy encoding in the H.264 main profile encoding. Given motion estimation and quantization step sizes, the proposed SDQ design provides near-optimal quantization outputs

[2]In this thesis, rate gain or rate reduction means the relative rate saving when PSNR is maintained the same.

for a given block in the sense of minimizing the actual RD cost when the adjacent block dependency is ignored. Experiment results show that our joint optimization encoder with the proposed SDQ based on CABAC achieves on average 10% rate reduction while maintaining the same video quality over the main-profile RD optimization method in [77] using CABAC, with half of the gain coming from the SDQ design. If compared with the baseline RD optimization method in [77], the proposed joint optimization encoder with SDQ based on CABAC achieves a 20% rate reduction, while the joint optimization encoder with SDQ based on CAVLC proposed in Chapter 4 achieves a 12% rate reduction. Overall, the proposed joint optimization encoder with SDQ based on CABAC shows the best coding performance that is known for H.264-compatible codecs.

Chapter 6

Image/Video Transcoding with Spatial Resolution Reduction

In this chapter, we investigate the trade-off between distortion and complexity for transcoding DCT images/video frames from a high spatial resolution to a low spatial resolution [82]. This is motivated by a desire to provide universal multimedia access over a network with diverse display devices. Specifically, we are focused on designing down-sampling algorithms in the DCT-domain, because most image/video data to be shared over the network are originally captured with high resolution and coded using a transform technique of DCT, e.g., MPEG, JPEG, DV, etc. In the following, we first review image down-sampling in the pixel domain. Then, we review related work for designing DCT-domain down-sampling methods in the literature. Finally, a designing framework is proposed for down-sampling DCT images/video with an arbitrary ratio.

6.1 Image Down-sampling in Pixel Domain

Image down-sampling in the pixel domain has been well studied and has become textbook material [60]. Consider a digital image $\boldsymbol{X}$ with sampling rate $f_{\text{s,high}}$. The problem of digital image down-sampling is to find a discrete image $\boldsymbol{x}$, which accords

to re-sampling $g(\boldsymbol{X})$ at a lower rate of $f_{\text{s,low}}$, where $g(\boldsymbol{X})$ denotes the continuous image which is to be reconstructed using $\boldsymbol{X}$. As a fundamental result in the field of information theory, Nyquist-Shannon sampling theorem states that the signal bandwidth must be no more than half of the sampling rate $\frac{1}{2}f_{\text{s}}$ in order to avoid aliasing. Hence, a low-pass filter is required to shrink the signal bandwidth of $g(\boldsymbol{X})$. As a result, the down-sampling includes two steps. First, a digital low-pass filter is used to process $\boldsymbol{X}$ to obtain $\hat{\boldsymbol{X}}$, whose bandwidth is less than $\frac{1}{2}f_{\text{s,low}}$. Second, the down-sampling is carried out by interpolating $\hat{\boldsymbol{X}}$ to compute the sample values at equally-spaced intervals given by $f_{\text{s,low}}$.

6.1.1 Low-pass Filtering for Down-sampling

In practice, there are several criteria that have been used for evaluating the low-pass filtering performance for the purpose of down-sampling. In the following, we summarize these criteria and use them to select a particular low-pass filter, based on which a DCT-domain down-sampling framework is constructed as to be presented later. Specifically, the following three criteria are investigated.

1. To remove aliases by quenching frequency components higher than the Nyquist frequency.

2. To limit the ringing effect by smoothing the transition band.

3. To keep the sharpness of the image by preserving as much energy as possible for frequency components lower than the Nyquist frequency.

By the Nyquist−Shannon sampling theorem, the main purpose of applying low-passing filtering is to remove aliases. An ideal low-pass filter may completely eliminate all frequencies above the Nyquist frequency while passing all those below, as shown in Figure 6.1. Furthermore, it can be proven that the ideal filter [60] gives the optimal performance of minimizing the L^2 distance between the original signal and its filtered version while completely removing aliases, as shown in the following.

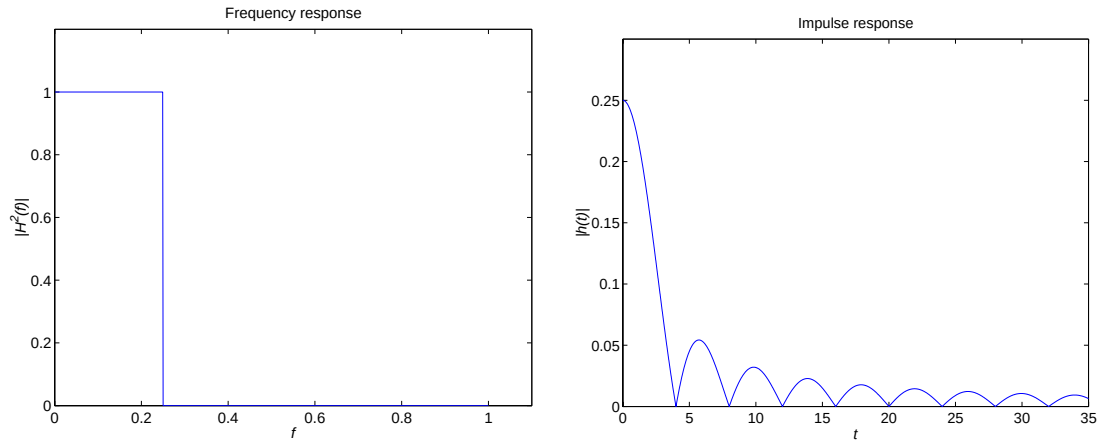


Figure 6.1: Frequency response and impulse response of an ideal low-pass filter with the cutoff frequency $f_0 = 0.2$.

Let $x(t)$ be any finite energy signal with Fourier transform $F(f)$. The ideal filter with its cutoff frequency being the Nyquist frequency results in the minimal L^2 distance between the the signal and the filtering output under the condition that there is no alias, i.e.,

$$H_{\text{ideal}}(f) = \arg\min_{H(f)} \int_{-\infty}^{\infty} |x(t) - \hat{x}(t)|^2 dt$$

where $\hat{x}(t) = \int_{-\infty}^{\infty} (F(f) \cdot H(f)) e^{j2\pi ft} df$. While this result is summarized for 1-dimensional signal, it can be extended to the case of filtering 2D data such as images. The proof can be done by using the Parseval's relation [60], i.e., $\int_{-\infty}^{\infty} |x(t)|^2 dt = \int_{-\infty}^{\infty} |F(f)|^2 df$.

In practice, however, the ideal filter is not desirable for low-pass filtering because it introduces a so-called ringing effect. Figure 6.2 shows the ringing effect by the ideal filter for processing an image with intensity edges. The reason is the sharp transition band of the filter. Figure 6.1 shows the shape of a one-dimensional ideal filter in both frequency and spatial domains. The sharp transition in the frequency domain corresponds to a long tail with multiple peaks in the spatial domain. By the convolution theorem, multiplication in the Fourier domain corresponds to a convolution in the spatial domain. The multiple peaks will produce unwanted ringing along intensity edges in the spatial domain when they are convoluted with the spatial signal.

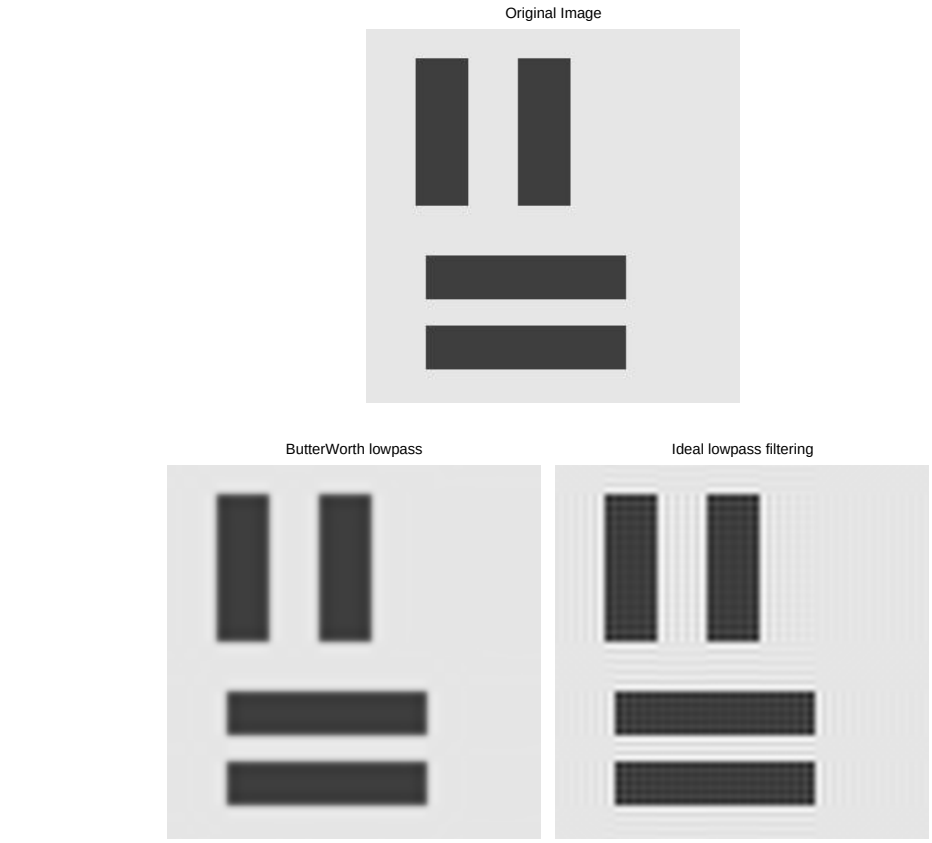


Figure 6.2: Low-pass filtering for an image with intensity edges. The cutoff frequency is 0.25. The PSNR by the Butterworth filter is 25.4dB, while the ideal filter results in a PSNR of 26.2dB. The ideal filtering result shows a clear ringing effect.

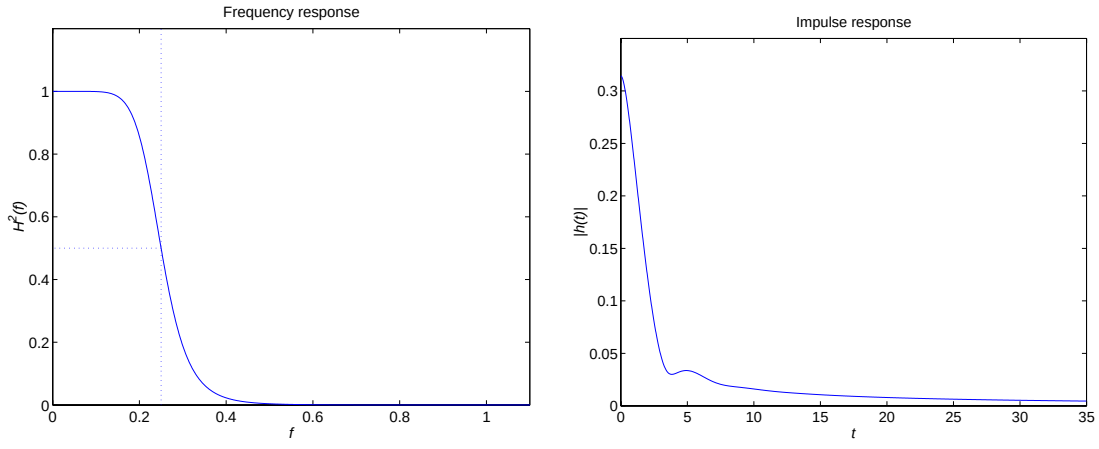


Figure 6.3: Frequency response and impulse response of a Butterworth filter.

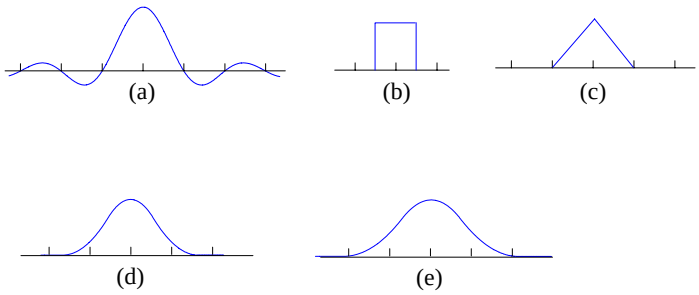


Figure 6.4: One-dimensional interpolation functions. (a) Sinc function for Nyquist−Shannon interpolation. (b) Square function for the nearest interpolation. (c) Triangle function (two squares convolved) for linear interpolation with a continuous but not smooth output. (d) Bell function (three squares convolved) for interpolation with continuous first order derivative. (e) Cubic B-spline function (four squares convolved) for interpolation with continuous second order derivative.

The ringing effect can be attenuated by smoothing the transition band of the filter. A Gaussian filter has a smooth Gaussian shape in both the frequency and spatial domains. It does not incur any ringing effect. However, the Gaussian shape in the frequency domain causes a significant loss of low frequency energy unnecessarily. As a result, the Gaussian filter is unsuitable for down-sampling. The Butterworth filter, however, provides a good solution for anti-aliasing and anti-ringing filtering due to its smooth transition band. As shown in Figure 6.3, it also preserves most low frequency energy. Indeed, simulation results, as to be shown later, show that down-sampling based on a Butterworth filter gives the best visual quality in comparison with another ideal-filter-based method and the 'resize' function in Matlab (A commercial software product from Mathworks) for down-sampling some benchmark images.

6.1.2 Interpolations

After low-pass filtering, image down-sampling becomes a problem of estimating the sample values at some points according to $f_{s,\text{low}}$ based on sample values at given

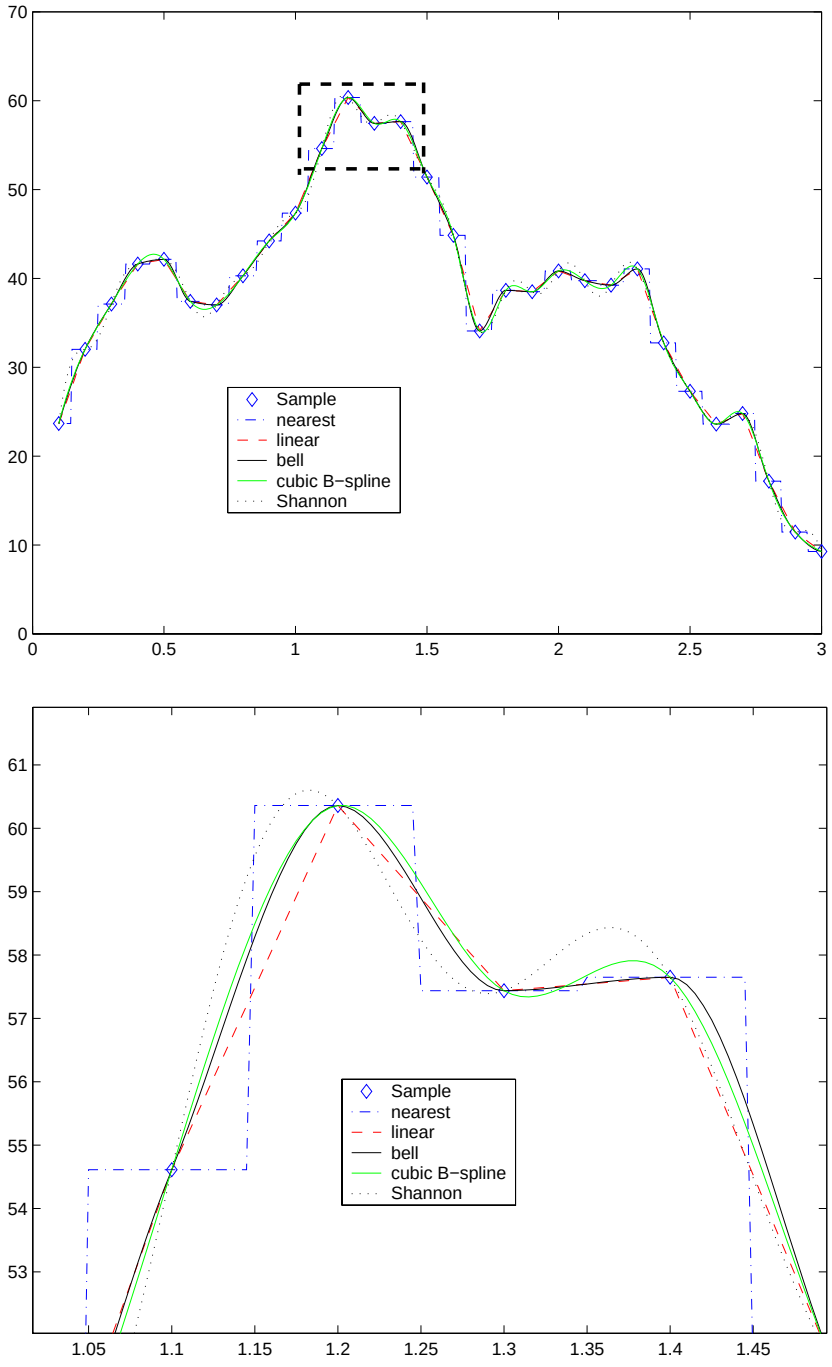


Figure 6.5: Demonstration of various interpolation methods. The lower panel shows the zoom-in of the square area in the upper figure.

points. In general, interpolation theory studies a problem of estimating the output of a function at arbitrary points based on its sample values at some given points[60]. The interpolation sometimes is also addressed as reconstruction filtering.

There have been a wide range of interpolation methods with various complexity and quality studied in the literature from the nearest neighbor interpolation to the Nyquist−Shannon interpolation. The nearest neighbor interpolation accords to a square function, while a better interpolation can be achieved by convolving several square functions to obtain the interpolation function. Figure 6.4 shows some one-dimensional interpolation functions. Interpolation in the spatial domain is to convolve the interpolation function with the pulse sequences of samples.

Figure 6.5 shows the outputs of 5 algorithms for interpolating some equally-spaced samples shown by diamonds. The nearest neighbor interpolation is the simplest one with the least complexity, yet it outputs a step function with discontinuous points. The linear interpolation outputs a continuous but not smooth function. The bell function interpolation provides a smooth output with continuous first order derivative. The cubic B-spline method, furthermore, generates an output with continuous second order derivative.

Theoretically, the Nyquist−Shannon method should give the best result for interpolating the samples with limited bandwidth. However, this theoretical result assumes an infinitely long sequence, which may have to be truncated in a practical system. The cubic B-spline interpolation method gives a solution with continuous second order derivative and generally provides a satisfying performance because it fits with a physical fact that the curvature of a curve at a point is determined by the second derivative at that point [60].

6.2 Review of DCT-domain Down-sampling methods

A straightforward method for down-sampling DCT images is to first convert DCT data back to the spatial domain and then apply a standard down-sampling method in the spatial domain. This method may give the best visual quality, yet its complexity is too high [52, 12, 58, 69]. As discussed previously, down-sampling in the spatial-domain consists of low-pass filtering and interpolation. Theoretically, low-pass filtering for down-sampling is justified by the Nyquist−Shannon sampling theorem for anti-aliasing. Technically, there have been many low-pass filter designs developed to further deal with practical issues such as the so-called ringing effect [44]. After low-pass filtering, down-sampling in the spatial domain becomes a problem of estimating sample values at certain points, for which the interpolation theory has been established. Practically, there have been a wide range of interpolation methods proposed in the literature with various complexity and quality, from the nearest neighbor interpolation to the spline interpolation [60]. Thus, it is fair to say that image down-sampling in the pixel domain has been well studied both in theory and in practice and it may give the best visual quality for image down-sampling. The problem for the above method, however, is the computational complexity[1], which is associated with the spatial-domain low-pass filtering, interpolation, as well as the inverse DCT and DCT.

In practice, a desirable down-sampling method for DCT data may consider three factors, i.e., the quality of the down-sampled image, the computational complexity, and the down-sampling ratio. The above method of transforming DCT data into spatial domain for down-sampling represents the best case in terms of quality, yet the worst case in terms of computational complexity. To tackle the complexity issue, it is desired that down-sampling is carried out in the DCT domain directly [5][48][12] without involving the inverse DCT of the original DCT data and the

[1]An experimental result for the complexity will be shown in Section 6.6.

subsequent DCT of the down-sampled data. Many methods along this line have been developed in the literature.

One category of DCT-domain down-sampling methods, referred to as DCT coefficient manipulation, investigate various properties of DCT coefficients and manipulate them with techniques such as zero-padding, truncating, scaling, etc. In [12], a fast algorithm was developed for down-sampling DCT images by a factor of 2 based on DCT coefficients truncation. In [56], based on studies of the symmetric convolution property of DCT, zero-padding and truncation were jointly utilized, leading to the so-called L/M-fold image resizing algorithm. The L/M-fold method was further accelerated in [58] by using fast algorithms for inverse and forward DCT transforms with composite lengths developed in the literature. In general, there is an inherent drawback for these manipulation methods, as also discussed in [12]. Specifically, the truncation of DCT coefficients is equivalent to an ideal filter with a sharp transition band in term of filtering. As discussed in Section 6.1, however, the ideal filter is not desirable because it introduces ringing effects [60].

Another category of such methods may be viewed as a linear transform of the DCT coefficients, which is equivalent to a concatenation of inverse DCT, a specific down-sampling method in the spatial domain, and DCT. In [52], a spatial-domain method of averaging each M×M block for down-sampling by a factor of M (M=2,3,4) was used to derive a fast down-sampling method in the DCT domain. The key idea was to derive a computationally efficient method for combining the inverse DCT, the spatial-domain down-sampling method, and the forward DCT into a one-stage computation to reduce the complexity. Later, this idea was extended to a general case with arbitrary down-sampling ratio in [48], where the so-called transform-domain resolution translation was developed based on a pipeline architecture that involves matrix-vector multiplications. For these methods, the quality is mainly determined by the corresponding method for down-sampling in spatial domain. The method by averaging in [52] overlooks the anti-aliasing filtering, resulting in a limited performance in term of quality. The scheme in [48] allows a flexible choice

of low-pass filtering. Hence, it is capable of achieving good quality for the obtained image. However, the computational complexity for matrix-vector multiplication is still relatively high [58].

6.3 Linear Transform with Double-sided Matrix Multiplication

This section derives a result that for a wide range of spatial-domain down-sampling methods, a concatenation of inverse DCT, spatial-domain down-sampling and DCT can be implemented equivalently as a linear transform with double-sided matrix multiplication (LTDS) in the DCT domain.

The derivation of LTDS may be summarized in 3 steps. Denote $\boldsymbol{t}$ as a DCT matrix. Consider to down-sample an M×N DCT image $\boldsymbol{C}_{\text{MN}}$ with a concatenation of inverse DCT, spatial-domain down-sampling and DCT. The 3 steps are as follows. First, apply the inverse DCT to obtain the spatial-domain image $\boldsymbol{X}_{\text{MN}}$ as

$$\boldsymbol{X}_{\text{MN}} = \boldsymbol{t}' \boxdot \boldsymbol{C}_{\text{MN}} \boxdot \boldsymbol{t}, \tag{6.1}$$

where $\boxdot$ denotes block-wise multiplications. Second, a spatial-domain method is selected and used to down-sample $\boldsymbol{X}_{\text{MN}}$ to obtain an $\text{I} \times \text{J}$ image, denoted as $\boldsymbol{x}_{\text{IJ}}$. Third, DCT is applied to the $\text{I} \times \text{J}$ image, resulting in

$$\boldsymbol{V}_{\text{IJ}} = \boldsymbol{t} \boxdot \boldsymbol{x}_{\text{IJ}} \boxdot \boldsymbol{t}'. \tag{6.2}$$

We now consider details of the second step, i.e., down-sampling in the spatial domain. Specifically, down-sampling in the spatial domain consists of low-pass filtering and interpolation. We consider to implement the low-pass filter based on a 2D discrete Fourier transform (DFT). Given the image $\boldsymbol{X}_{\text{MN}}$, the filtering output $\tilde{\boldsymbol{X}}_{\text{MN}}$ is obtained by

$$\tilde{\boldsymbol{X}}_{\text{MN}} = \boldsymbol{A}^*_{\text{MM}}((\boldsymbol{A}_{\text{MM}} \cdot \boldsymbol{X}_{\text{MN}} \cdot \boldsymbol{B}_{\text{NN}}) \otimes \boldsymbol{F}_{\text{MN}}) \cdot \boldsymbol{B}^*_{\text{NN}}, \tag{6.3}$$

where $\boldsymbol{A}_{\text{MM}}$ is an $\text{M} \times \text{M}$ DFT transform matrix with its element given by

$$a_{uv} = \frac{1}{\sqrt{\mathrm{M}}} \exp\left(\frac{-j2\pi uv}{\mathrm{M}}\right), \qquad u = 0, 1, \cdots, \mathrm{M}-1, \quad v = 0, 1, \cdots, \mathrm{M}-1,$$

and $\boldsymbol{A}^*_{\mathrm{MM}}$ is its conjugate. Similarly, $\boldsymbol{B}_{\mathrm{NN}}$ is an NxN DFT transform matrix and $\boldsymbol{B}^*_{\mathrm{NN}}$ is the conjugate matrix. $\boldsymbol{F}_{\mathrm{MN}}$ is the low-pass filtering matrix in the DFT domain. The symbol $\otimes$ denotes element-wise multiplications.

Consider to construct $\boldsymbol{F}_{\mathrm{MN}}$ based on two one-dimensional filters[2], i.e.,

$$\boldsymbol{F}_{\mathrm{MN}} = L_{\mathrm{M1}} \cdot R_{\mathrm{1N}}. \tag{6.4}$$

It is then not hard to see that the element wise multiplication in (6.3) may be removed, yielding

$$\tilde{\boldsymbol{X}}_{\mathrm{MN}} = \boldsymbol{A}^*_{\mathrm{MM}} \cdot \boldsymbol{L}_{\mathrm{MM}} \cdot (\boldsymbol{A}_{\mathrm{MM}} \cdot \boldsymbol{X}_{\mathrm{MN}} \cdot \boldsymbol{B}_{\mathrm{NN}}) \cdot \boldsymbol{R}_{\mathrm{NN}} \cdot \boldsymbol{B}^*_{\mathrm{NN}}, \tag{6.5}$$

where $\boldsymbol{L}_{\mathrm{MM}}$ and $\boldsymbol{R}_{\mathrm{NN}}$ are diagonal matrixes with diagonal elements being L_{M1} and R_{1N}, respectively.

The interpolation is a process of reconstructing any in-between samples from the original samples, which is usually implemented as an interpolation filter. Assume a linear interpolation filter in the form of matrix multiplication. Specifically, denote $\boldsymbol{E}_{\mathrm{IM}}$ and $\boldsymbol{G}_{\mathrm{NJ}}$ as the interpolation matrixes. The down-sampled image is computed by

$$\boldsymbol{x}_{\mathrm{IJ}} = \boldsymbol{E}_{\mathrm{IM}} \cdot \tilde{\boldsymbol{X}}_{\mathrm{MN}} \cdot \boldsymbol{G}_{\mathrm{NJ}}. \tag{6.6}$$

Insert (6.5) into (6.6). Down-sampling in the spatial domain is carried out by

$$\boldsymbol{x}_{\mathrm{IJ}} = \boldsymbol{E}_{\mathrm{IM}} \boldsymbol{A}^*_{\mathrm{MM}} \boldsymbol{L}_{\mathrm{MM}} \boldsymbol{A}_{\mathrm{MM}} \cdot \boldsymbol{X}_{\mathrm{MN}} \cdot \boldsymbol{B}_{\mathrm{NN}} \boldsymbol{R}_{\mathrm{NN}} \boldsymbol{B}^*_{\mathrm{NN}} \boldsymbol{G}_{\mathrm{NJ}}. \tag{6.7}$$

Now, combine (6.1), (6.2) and (6.7). The concatenation of inverse DCT, spatial-domain down-sampling, and DCT is,

$$\boldsymbol{V}_{\mathrm{IJ}} = \boldsymbol{t} \boxdot [\boldsymbol{E}_{\mathrm{IM}} \boldsymbol{A}^*_{\mathrm{MM}} \boldsymbol{L}_{\mathrm{MM}} \boldsymbol{A}_{\mathrm{MM}} (\boldsymbol{t}' \boxdot \boldsymbol{C}_{\mathrm{MN}} \boxdot \boldsymbol{t}) \cdot \boldsymbol{B}_{\mathrm{NN}} \boldsymbol{R}_{\mathrm{NN}} \boldsymbol{B}^*_{\mathrm{NN}} \boldsymbol{G}_{\mathrm{NJ}}] \boxdot \boldsymbol{t}'$$

[2]Because the 2D filtering may be implemented with a concatenation of two 1D filters, the 2D filtering matrix $\boldsymbol{F}_{\mathrm{MN}}$ is assumed to take the form of $L_{\mathrm{M1}} \cdot R_{\mathrm{1N}}$. This, in turn, helps to replace the element-wise multiplication in (6.3) with regular matrix multiplication, which is a necessary condition for obtaining the LTDS.

The block-wise multiplication can be replaced by applying a result of $\boldsymbol{t} \boxdot \boldsymbol{C}_{\text{MN}} = \boldsymbol{T}_{\text{MM},\boldsymbol{t}} \cdot \boldsymbol{C}_{\text{MN}}$, where $\boldsymbol{T}_{\text{MM},\boldsymbol{t}} = \begin{pmatrix} \lfloor \boldsymbol{t} \rfloor & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \ddots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \lfloor \boldsymbol{t} \rfloor \end{pmatrix}$ with the DCT matrix $\boldsymbol{t}$ lining on the diagonal of $\boldsymbol{T}$. Consequently, we obtain a linear transform in the DCT domain,

$$\boldsymbol{V}_{\text{IJ}} = \boldsymbol{D}_{\text{IM}} \cdot \boldsymbol{C}_{\text{MN}} \cdot \boldsymbol{W}_{\text{NJ}}, \tag{6.8}$$

where

$$\begin{aligned} \boldsymbol{D}_{\text{IM}} &= \boldsymbol{T}_{\text{II},\boldsymbol{t}} \boldsymbol{E}_{\text{IM}} \boldsymbol{A}^{*}_{\text{MM}} \boldsymbol{L}_{\text{MM}} \boldsymbol{A}_{\text{MM}} \boldsymbol{T}_{\text{MM},\boldsymbol{t}'}, \\ \boldsymbol{W}_{\text{NJ}} &= \boldsymbol{T}_{\text{NN},\boldsymbol{t}} \boldsymbol{B}_{\text{NN}} \boldsymbol{R}^{*}_{\text{NN}} \boldsymbol{B}_{\text{NN}} \boldsymbol{G}_{\text{NJ}} \boldsymbol{T}_{\text{JJ},\boldsymbol{t}'}. \end{aligned}$$

Equation (6.8) shows that for many down-sampling methods with a concatenation of inverse DCT, spatial-domain down-sampling, and DCT, we can find an equivalent linear transform in the DCT domain, which produces the same output. Motivated by (6.8), we call any DCT domain transform in the form of $\boldsymbol{D}_{\text{IM}} \cdot \boldsymbol{C}_{\text{MN}} \cdot \boldsymbol{W}_{\text{NJ}}$ a DCT-domain LTDS, where $\boldsymbol{D}_{\text{IM}}$ and $\boldsymbol{W}_{\text{NJ}}$ are arbitrary matrixes with respective dimensions. We are interested in the set of all DCT-domain LTDSs, denoted as S hereafter. Clearly, as shown in (6.8), the set is quite large; It contains all methods corresponding to spatial-domain down-sampling methods with properties of (6.4) and (6.6). In particular, it includes the L/M-fold method (since the truncation operation of DCT coefficient in the L/M-fold method is equivalent to a filtering process with properties of (6.4) and (6.6)), as well as the methods in [52] and [48]. Some LTDSs in the set have high complexity while others have low complexity. Given any spatial-domain method which may not be in the set S, it will be interesting to find its LTDS approximation in S, which gives the best trade-off between the visual quality and the computational complexity. In the following, we will propose a framework for designing LTDSs corresponding to a given spatial-domain down-sampling method by jointly optimizing the visual quality and the computational complexity.

6.4 Visual Quality Measurement for Down-sampled Images

Quality measure is the very basis to formulate our optimization problem for finding optimal LTDSs corresponding to a pre-selected spatial-domain down-sampling method. Although spatial-domain down-sampling has become textbook material for decades [44], unfortunately, there is still no objective measurement unanimously accepted for measuring the quality of a down-sampled image.

There have been two major objective quality measures used in the literature for image down-sampling. The first one is to measure the quality with a reference image obtained using a standard down-sampling method in the spatial domain [52][53]. The second one is to up-sample the down-sampled image to the original resolution. Then, the quality is measured by the MSE between the up-sampled image and the original one [12, 56, 57].

In this research, we apply the first measure for evaluating the performance of different LTDSs, based on which an optimization problem is formulated to find an LTDS to achieve the best trade-off between the visual quality and the complexity. This measurement will naturally allow us to approach the visual performance of a pre-selected spatial-domain method by setting the down-sampling output of the pre-selected method as the reference. Then, we may be able to achieve the best visual quality by pre-selecting a spatial-domain method with the best visual quality if there is one. Meanwhile, since we are interested in viewing a down-sampled image in its own resolution, there is no up-sampling process involved. Hence the first measure is more appropriate than the second measure for our purpose. Besides, the second measure is contradict with the principal of anti-aliasing filtering for down-sampling in the sense that an optimization problem for minimizing the MSE between the up-sampled image and the original one will treat the anti-aliasing filtering as a source of information loss for high-frequency components and will tend to minimize such loss. In addition, observations in [12] also show that the MSE between the up-

sampled image and the original one is more determined by the up-sampling scheme than by the down-sampling method. Because we target a down-sampling design and the up-sampling process is out of the scope of our optimization, the second measure is unsuitable in our design.

Specifically, the quality measure used in our problem formulation is as follows. Consider an M×N DCT image $\boldsymbol{C}_{\mathrm{MN}}$. The original image in the spatial domain is $\boldsymbol{X}_{\mathrm{MN}} = \mathrm{T}^{-1}(\boldsymbol{C}_{\mathrm{MN}})$, where $\mathrm{T}^{-1}(\cdot)$ represents the inverse DCT. Assume $\boldsymbol{x}_{\mathrm{IJ}}$ as the I×J reference image obtained using a pre-selected spatial-domain method for down-sampling $\boldsymbol{X}_{\mathrm{MN}}$. For any LTDS in S, the quality of the obtained image $\boldsymbol{D}_{\mathrm{IM}} \cdot \boldsymbol{C}_{\mathrm{MN}} \cdot \boldsymbol{W}_{\mathrm{NJ}}$ is measured by $||\boldsymbol{x}_{\mathrm{IJ}} - \mathrm{T}^{-1}(\boldsymbol{D}_{\mathrm{IM}} \cdot \boldsymbol{C}_{\mathrm{MN}} \cdot \boldsymbol{W}_{\mathrm{NJ}})||^2$.

In complement to the objective measure discussed in the above, we will also use subjective evaluation and present resulting images in the experimental section, since the down-sampling output eventually go to a human viewer. The subjective visual quality of a down-sampling image is mainly evaluated by appearance of aliasing, ringing effect, and/or other artifacts.

6.5 LTDS-based Down-sampling Design

6.5.1 Complexity Modeling of LTDS

In general, the complexity for computing $\boldsymbol{D}_{\mathrm{IM}} \cdot \boldsymbol{C}_{\mathrm{MN}} \cdot \boldsymbol{W}_{\mathrm{NJ}}$ is related to the number of non-zero elements in $\boldsymbol{D}_{\mathrm{IM}}$ and $\boldsymbol{W}_{\mathrm{NJ}}$. Specifically, the computation of $\boldsymbol{D}_{\mathrm{IM}} \cdot \boldsymbol{C}_{\mathrm{MN}} \cdot \boldsymbol{W}_{\mathrm{NJ}}$, if computed from left to right, involves p_1·I·M·N+p_2·I·N·J multiplications and I·(p_1M − 1)·N+I·(p_2N − 1)·J additions, where p_1 and p_2 are the percentage of nonzero elements in $\boldsymbol{D}_{\mathrm{IM}}$ and $\boldsymbol{W}_{\mathrm{NJ}}$, respectively. To reduce the computational complexity, we plan to apply a structural learning with forgetting (SLF) scheme[43] to decrease p_1 and p_2. Initially, we consider a complexity model of

$$r_{\mathrm{f}} = |\boldsymbol{D}_{\mathrm{IM}}| + |\boldsymbol{W}_{\mathrm{NJ}}|,$$

where $|\cdot|$ defines the l_1 norm of a matrix. By a learning with forgetting stage of SLF, the minimization of $|\boldsymbol{D}_{\text{IM}}| + |\boldsymbol{W}_{\text{NJ}}|$ will lead to a constant decay for all non-zero elements, forcing as many elements to be zero as possible. This constant decay at the beginning stage also helps to remove redundant connections due to a random initialization that is typically adopted before SLF is applied.

The complexity model needs to be further adjusted for a learning with selective forgetting stage in SLF, which follows the learning with forgetting stage. In general, a constant decay to elements with large values will introduce a large distortion to the visual quality, measured as $||\boldsymbol{x}_{\text{IJ}} - \text{T}^{-1}(\boldsymbol{D}_{\text{IM}} \cdot \boldsymbol{C}_{\text{MN}} \cdot \boldsymbol{W}_{\text{NJ}})||^2$. After removing redundant connections due to a random initialization of all elements, we expect to protect certain large elements from the decay so that they can be trained to focus on providing better visual quality. Accordingly, the complexity model for the learning with selective forgetting stage is defined as

$$r_{\text{s}} = |\boldsymbol{D}_{\text{IM}}|_{|d_{im}|<d_0} + |\boldsymbol{W}_{\text{NJ}}|_{|w_{nj}|<w_0},$$

where d_0 and w_0 are two thresholds, and $|\boldsymbol{D}_{\text{IM}}|_{|d_{im}|<d_0}$ ($|\boldsymbol{W}_{\text{NJ}}|_{|w_{nj}|<w_0}$, respectively) denotes the modified l_1 norm of $\boldsymbol{D}_{\text{IM}}$ ($\boldsymbol{W}_{\text{NJ}}$, respectively) in which all elements of $\boldsymbol{D}_{\text{IM}}$ ($\boldsymbol{W}_{\text{NJ}}$, respectively) with magnitude greater than or equal to d_0 (w_0, respectively) are excluded. The minimization of this complexity function will lead to a constant decay only to elements with small values and will force them to zero, while elements with large values are excluded from the complexity model.

Besides the number of non-zero elements in $\boldsymbol{D}_{\text{IM}}$ and $\boldsymbol{W}_{\text{NJ}}$, the complexity for computing $\boldsymbol{D}_{\text{IM}} \cdot \boldsymbol{C}_{\text{MN}} \cdot \boldsymbol{W}_{\text{NJ}}$ is also related to how multiplications may be implemented. In general, a multiplication may be approximated by a series of additions and shifts, e.g., for a multiplier $\mathsf{a} \backsimeq \sum a_i \cdot 2^{-i}, \quad a_i \in \{1, -1, 0\}$, we have $\mathsf{a} \cdot v \backsimeq \sum a_i \cdot (v >> i)$, where a_i determines the sign and '>>' stands for right shift. This approximation is desirable if the quality loss due to the resulted inaccuracy and the complexity reduction due to the fast implementation of shifts and additions are well balanced. Assuming the magnitudes of all elements in $\boldsymbol{D}_{\text{IM}}$ and $\boldsymbol{W}_{\text{NJ}}$ are

in the range of $[0, 8)$[3], we introduce the following quantization procedure into the complexity model. For any $x \in (-8, 8)$, define

$$\mathrm{Q}(x) = \sum_{i=-2}^{i=15} a_i \cdot 2^{-i}, \quad a_i \in \{1, -1, 0\} \tag{6.9}$$

where

$$\{a_i\} = \arg \min_{|x - \sum(a_i 2^{-i})| \leq |x|\eta} \sum |a_i|,$$

where η is a small constant. Essentially, this quantization procedure leads to an approximation within a given neighboring region with the minimal number of ones in the binary representation. Thus, the corresponding multiplication may be implemented with the minimal number of shifts and additions.

The quantization procedure discussed above is generally applied at the learning with selective forgetting stage of SLF because its corresponding contribution to the complexity function is at a level similar to r_{s}, which is much less than r_{f}. Overall, the complexity model for the learning with selective forgetting stage is defined as follows,

$$r_{\mathrm{q}} = (|\boldsymbol{D}_{\mathrm{IM}}|_{|d_{im}|<d_0} + |\boldsymbol{W}_{\mathrm{NJ}}|_{|w_{nj}|<w_0}) + \rho \cdot (|\boldsymbol{D}_{\mathrm{IM}} - Q(\boldsymbol{D}_{\mathrm{IM}})| + |\boldsymbol{W}_{\mathrm{NJ}} - Q(\boldsymbol{W}_{\mathrm{NJ}})|), \tag{6.10}$$

where ρ is a constant, and $Q(\boldsymbol{D}_{\mathrm{IM}})$ and $Q(\boldsymbol{W}_{\mathrm{NJ}})$ mean to apply $Q(\cdot)$ to each element of $\boldsymbol{D}_{\mathrm{IM}}$ and $\boldsymbol{W}_{\mathrm{NJ}}$.

6.5.2 Optimization Problem Formulation

Based on the above discussions on LTDS and its complexity models, we now formulate the design problem for down-sampling in the DCT domain as a joint optimization of the visual quality and the computational complexity, i.e.,

$$\min_{g(\cdot)} ||g(\boldsymbol{C}_{\mathrm{MN}}) - \boldsymbol{V}_{\mathrm{IJ}}||^2 + \lambda \cdot r_g, \tag{6.11}$$

[3]This range is set up empirically as observation shows that almost all elements in $\boldsymbol{D}_{\mathrm{IM}}$ and $\boldsymbol{W}_{\mathrm{NJ}}$ have magnitudes strictly smaller than 1.

where $\boldsymbol{C}_{\text{MN}}$ is a DCT image and $\boldsymbol{V}_{\text{IJ}}$ is the down-sampling output for $\boldsymbol{C}_{\text{MN}}$ using a pre-selected spatial-domain method. Consider $g(\cdot)$ as the LTDS in (6.8). The complexity term r_g may take the definition of r_{f} or r_{q} according to different stages of SLF. The optimization problem of (6.11) becomes

$$\min_{\boldsymbol{D}_{\text{IM}}, \boldsymbol{W}_{\text{NJ}}} ||\boldsymbol{D}_{\text{IM}} \boldsymbol{C}_{\text{MN}} \boldsymbol{W}_{\text{NJ}} - \boldsymbol{V}_{\text{IJ}}||^2 + \lambda \cdot r_g. \tag{6.12}$$

The objective of designing down-sampling algorithms in the DCT domain is to find an LTDS with the best trade-off between the fidelity of $g(\boldsymbol{C}_{\text{MN}})$ to $\boldsymbol{V}_{\text{IJ}}$ and the complexity of r_g in the sense of minimizing the joint cost. Note that the trade-off depends on five parameters, η, λ, ρ, d_0 and w_0, which are to be determined according to user preferences.

The above optimization problem involves a pre-selected down-sampling method. However, the problem formulation and the algorithms to be discussed later for solving the problem do not depend on any selected method. Instead, the optimization framework and the proposed learning algorithms for solving the problem take the output of a pre-selected method to form the training data. As to be presented later, experiments in this work are based on a spatial-domain method with Butterworth filter and cubic B-spline interpolation, which, according to our literature survey, is a popular choice for its advantages of anti-aliasing, ringing avoidance, and low-frequency components preservation. Yet, the proposed framework itself does not rely on this selection and it can be used to match any other spatial-domain method in the DCT domain.

6.5.3 Problem Solution

The optimization problem (6.12) is solved by modeling LTDS as a multiple-layer neural network. A structural learning with forgetting algorithm [43] is then used to train the network structure.

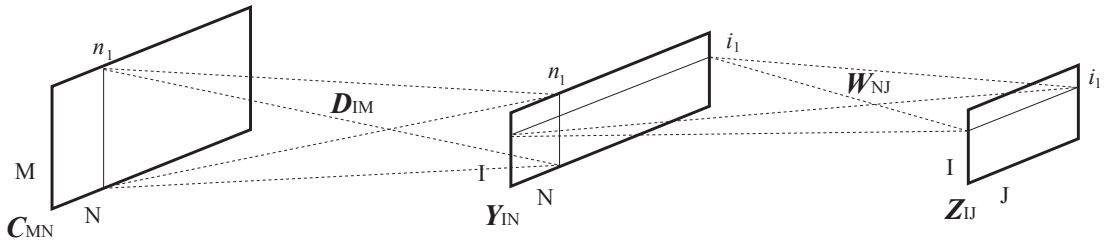


Figure 6.6: A three-layer network for implementing the linear transform of (6.8).

A Multiple-Layer Neural Network Structure

As shown in Figure 6.6, an LTDS may be implemented as a three-layer neural network. Similar to the multiple layer perceptron (MLP) [50], the three layers are named as input layer $\boldsymbol{C}_{\text{MN}}$, hidden layer $\boldsymbol{Y}_{\text{IN}}$, and output layer $\boldsymbol{Z}_{\text{IJ}}$. Then, connections are selectively built up among units in each two layers to simulate the matrix multiplication operation in the linear transform.

The left panel of Figure 6.7 shows the connections between the input layer and the hidden layer. Specifically, these connections are established according to three rules, i.e.,

- Connections are established from units in a given column of the input layer to units in the same column of the hidden layer. Note that the input layer and the hidden layer have the same number of columns.
- Units in a given column of the input layer are fully connected to units in the same column of the hidden layer.
- Valid connections between any two columns share the same weight matrix, i.e., $\boldsymbol{D}_{\text{IM}}$.

Consequently, the output of the hidden layer is computed as $\boldsymbol{Y}_{\text{IN}} = \boldsymbol{D}_{\text{IM}} \cdot \boldsymbol{C}_{\text{MN}}$ by a forward process from the input layer to the hidden layer.

Similarly, connections between the hidden layer and the output layer are demonstrated in the right panel of Figure 6.7. The connections rules are the same as the above except that connections are built up among rows and the corresponding

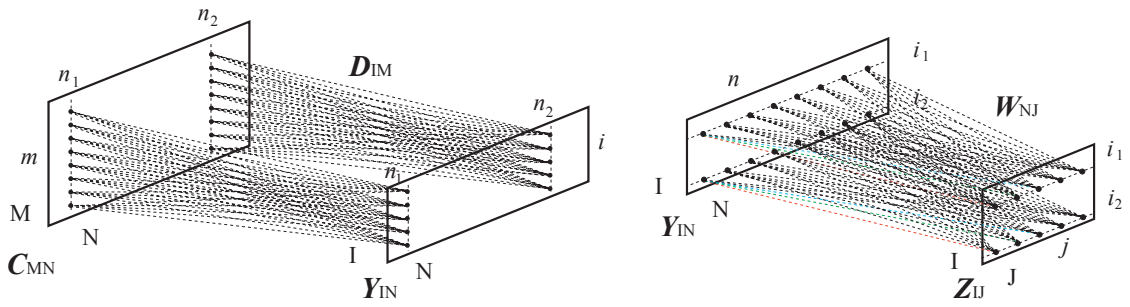


Figure 6.7: Illustration of selective connections in a three-layer network structure for simulating the computation of LTDS. The left panel shows connections between the input layer $\boldsymbol{C}_{\text{MN}}$ and the hidden layer $\boldsymbol{Y}_{\text{IN}}$. The right panel demonstrates connections between the hidden layer $\boldsymbol{Y}_{\text{IN}}$ and the output layer $\boldsymbol{Z}_{\text{IJ}}$.

weight matrix is $\boldsymbol{W}_{\text{NJ}}$. Then, forwarding computation from the hidden layer to the output layer leads to $\boldsymbol{Z}_{\text{IJ}} = \boldsymbol{Y}_{\text{IN}} \cdot \boldsymbol{W}_{\text{NJ}}$. Overall, the LTDS is implemented by a forwarding computation in the network structure as $\boldsymbol{Z}_{\text{IJ}} = \boldsymbol{D}_{\text{IM}} \cdot \boldsymbol{C}_{\text{MN}} \cdot \boldsymbol{W}_{\text{NJ}}$.

Training with Structural Learning with Forgetting

SLF was originally developed in [43] to find a concise structure for multiple-layer neural networks. The key idea of SLF is to simplify and clarify the network structure by removing redundant connections with a decay. In this study, SLF is adopted to reduce connections in the 3-layer network, so as to reduce the computation complexity of LTDS. Specifically, the learning procedure includes two stages, i.e., learning with forgetting and learning with selective forgetting.

The learning with forgetting stage is developed to remove redundant initial connections as much as possible. In this stage, the learning objective function is obtained by plugging $r_g = r_{\text{f}}$ into the objective function of (6.12), i.e.,

$$J_f = ||\boldsymbol{D}_{\text{IM}} \boldsymbol{C}_{\text{MN}} \boldsymbol{W}_{\text{NJ}} - \boldsymbol{V}_{\text{IJ}}||^2 + \lambda \cdot r_{\text{f}}. \tag{6.13}$$

Due to a random initialization of the connection weights, some redundant connections may possess an initial weight with a big value. Thus, it is desired to apply a

constant decay to all elements in order to get rid of this redundancy. Specifically, the learning with forgetting procedure is as follows,

1. Pass the input signal forward to compute the network outputs.

$$\boldsymbol{Y}_{\text{IN}} = \boldsymbol{D}_{\text{IM}} \cdot \boldsymbol{C}_{\text{MN}} \quad \Rightarrow \quad \boldsymbol{Z}_{\text{IJ}} = \boldsymbol{Y}_{\text{IN}} \cdot \boldsymbol{W}_{\text{NJ}}$$

2. Compute the network error and propagate it backward.

$$\Delta \boldsymbol{Z}_{\text{IJ}} = \boldsymbol{Z}_{\text{IJ}} - \boldsymbol{V}_{\text{IJ}} \quad \Rightarrow \quad (\Delta \boldsymbol{Y})_{\text{IN}} = (\Delta \boldsymbol{Z})_{\text{IJ}} \cdot (\boldsymbol{W}^{\text{t}})_{\text{JN}}$$

3. Compute the learning amount for $\boldsymbol{D}$ and $\boldsymbol{W}$.

$$\begin{array}{rcl} \Delta \boldsymbol{D} & = & \frac{1}{2} \cdot \frac{\partial J_f}{\partial \boldsymbol{D}} \quad = \quad (\Delta \boldsymbol{Y})_{\text{IN}} \cdot (\boldsymbol{C}^{\text{t}})_{\text{NM}} + \lambda \cdot \text{sgn}(\boldsymbol{D}_{\text{IM}}) \\ \Delta \boldsymbol{W} & = & \frac{1}{2} \cdot \frac{\partial J_f}{\partial \boldsymbol{W}} \quad = \quad (\boldsymbol{Y}^{\text{t}})_{\text{NI}} \cdot (\Delta \boldsymbol{Z})_{\text{IJ}} + \lambda \cdot \text{sgn}(\boldsymbol{W}_{\text{NJ}}) \end{array} \tag{6.14}$$

   where $\text{sgn}(\cdot)$ is the sign function as

$$\text{sgn}(x) = \begin{cases} \frac{1}{2}, & x > 0 \\ 0, & x = 0 \\ -\frac{1}{2}, & x < 0 \end{cases} .$$

4. Learn with error propagation and forgetting.

$$\begin{array}{rcl} \boldsymbol{D}_{\text{NJ}}^{(n+1)} & = & \boldsymbol{D}_{\text{NJ}}^{(n)} - \alpha \cdot \Delta \boldsymbol{D}, \\ \boldsymbol{W}_{\text{NJ}}^{(n+1)} & = & \boldsymbol{W}_{\text{NJ}}^{(n)} - \alpha \cdot \Delta \boldsymbol{W}, \end{array}$$

   where α is a small positive number named the learning factor and the superscripts (n) and $(n+1)$ accord to the n-th and $(n+1)$-th iterations. Note that the superscripts are omitted in steps 1 to 4 for simplicity.

5. Repeat steps 1 to 4 until the decrement of J_f is smaller than a given threshold.

The above learning with forgetting stage normally ends with a skeleton structure but a large distortion. The selective forgetting stage is then used to tune the structure for a better trade-off between distortion and complexity. Specifically, the selective forgetting stage accords to using the complexity model of r_{q} into the

minimization objective function. Apparently, a small threshold is introduced for selectively applying the decay to connections with small weights. Compared with the learning with forgetting stage discussed in the above, the algorithm for the selective forgetting is mostly the same, except the computation of the learning amount $\Delta \boldsymbol{D}$ and $\Delta \boldsymbol{W}$. For the selective forgetting stage, the learning amount is obtained by

$$\begin{array}{rcl} \Delta \boldsymbol{D} & = & (\Delta \boldsymbol{Y})_{\mathrm{IN}} \cdot (\boldsymbol{C}^{\mathrm{t}})_{\mathrm{NM}} + \lambda \cdot \mathrm{thr}(\boldsymbol{D}_{\mathrm{IM}}, d_0) + \lambda \cdot \rho \cdot \mathrm{sgn}(\boldsymbol{D}_{\mathrm{IM}} - Q(\boldsymbol{D}_{\mathrm{IM}})) \\ \Delta \boldsymbol{W} & = & (\boldsymbol{Y}^{\mathrm{t}})_{\mathrm{NI}} \cdot (\Delta \boldsymbol{Z})_{\mathrm{IJ}} + \lambda \cdot \mathrm{thr}(\boldsymbol{W}_{\mathrm{NJ}}, w_0) + \lambda \cdot \rho \cdot \mathrm{sgn}(\boldsymbol{W}_{\mathrm{NJ}} - Q(\boldsymbol{W}_{\mathrm{NJ}})) \end{array} ,$$

where ρ and $Q(\cdot)$ are defined in (6.10), and $\mathrm{thr}(\cdot)$ is defined as follows,

$$\mathrm{thr}(x, \theta) = \begin{cases} \frac{1}{2}, & \theta > x > 0 \\ 0, & x = 0 \ \text{ or } \ x \geq \theta \ \text{ or } \ x \leq -\theta \\ -\frac{1}{2}, & -\theta < x < 0 \end{cases} .$$

Efficient Down-sampling Algorithm Design

Based on the 3-layer structure and the structural learning with forgetting algorithm, the optimization problem in (6.12) is solved as follows,

1. Generate a training set based on a given spatial-domain down-sampling method which down-samples an M×N image to a resolution of I×J. Choose several M×N DCT images, $\{\boldsymbol{C}_{\mathrm{MN},i}, i = 1, \cdots, 5\}$. Apply the pre-selected down-sampling method discussed in Section 6.5.3 to obtain down-sampling references $\{\boldsymbol{V}_{\mathrm{IJ},i}, i = 1, \cdots, 5\}$. The training set is $\{(\boldsymbol{C}_{\mathrm{MN},i}, \boldsymbol{V}_{\mathrm{IJ},i}), i = 1, \cdots, 5\}$.

2. Learning with forgetting. Construct the 3-layer structure with $\boldsymbol{D}_{\mathrm{IM}}$ and $\boldsymbol{W}_{\mathrm{NJ}}$. Find a skeleton structure using the learning with forgetting algorithm.

3. Learning with selective forgetting. Refine $\boldsymbol{D}_{\mathrm{IM}}$ and $\boldsymbol{W}_{\mathrm{NJ}}$ with the learning with selective forgetting algorithm.

4. Combination of arithmetic operations to further reduce the computation cost.

The above algorithm results in an LTDS-based down-sampling method, which minimizes the joint cost of visual quality and complexity for given parameters of λ, η, ρ, d_0 and w_0. Essentially, different parameters will lead to a method with various complexity and different visual quality as well. End users may determine values for these parameters according to their requirements on the acceptable quality and the affordable complexity.

Performance Analysis

Convergence of the Learning Algorithm. In general, the global convergence of SLF for solving (6.12) is not guaranteed. Still, we may show that the learning with forgetting algorithm will converge for minimizing (6.13) based on a given pair of training data ($\boldsymbol{C}_{\mathrm{MN},i}$, $\boldsymbol{V}_{\mathrm{IJ},i}$). Consider the Hessian matrix corresponding to $\boldsymbol{W}_{\mathrm{NJ}}$.

$$\boldsymbol{G}_{\mathrm{NJ}\times\mathrm{NJ}}(\boldsymbol{W}) = \begin{pmatrix} \frac{\partial J_f}{\partial w_{11}\partial w_{11}} & \frac{\partial J_f}{\partial w_{11}\partial w_{12}} & \cdots & \frac{\partial J_f}{\partial w_{11}\partial w_{\mathrm{NJ}}} \\ \vdots & \cdot & \ddots & \vdots \\ \frac{\partial J_f}{\partial w_{NJ}\partial w_{11}} & \frac{\partial J_f}{\partial w_{NJ}\partial w_{12}} & \cdots & \frac{\partial J_f}{\partial w_{NJ}\partial w_{\mathrm{NJ}}} \end{pmatrix}.$$

By some derivation, we have

$$\boldsymbol{G}_{\mathrm{NJ}\times\mathrm{NJ}}(\boldsymbol{W}) = \begin{pmatrix} [\![\mathcal{G}_{\mathrm{JJ}}]\!]_1 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \ddots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & [\![\mathcal{G}_{\mathrm{JJ}}]\!]_{\mathrm{N}} \end{pmatrix},$$

with matrixes $\mathcal{G}_{\mathrm{JJ}}$ lying on the diagonal and $\mathcal{G}_{\mathrm{JJ}} = (\Delta\boldsymbol{Z}^{\mathrm{t}})_{\mathrm{JI}} \cdot \Delta\boldsymbol{Z}_{\mathrm{IJ}}$. Apparently, $\mathcal{G}_{\mathrm{JJ}}$ is positive semi-definite. Therefore, the Hessian matrix $\boldsymbol{G}_{\mathrm{NJ}\times\mathrm{NJ}}(\boldsymbol{W})$ is positive semi-definite. Similarly, we can show that the Hessian matrix corresponding to $\boldsymbol{D}_{\mathrm{IM}}$ is,

$$\boldsymbol{H}_{\mathrm{IM}\times\mathrm{IM}}(\boldsymbol{D}) = \begin{pmatrix} [\![\mathcal{H}_{\mathrm{MM}}]\!]_1 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \ddots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & [\![\mathcal{H}_{\mathrm{MM}}]\!]_{\mathrm{I}} \end{pmatrix},$$

with $\mathcal{H}_{\mathrm{MM}} = \boldsymbol{C}_{\mathrm{MN}} \cdot \boldsymbol{W}_{\mathrm{NJ}} \cdot (\boldsymbol{W}^{\mathrm{t}})_{\mathrm{JN}} \cdot (\boldsymbol{C}^{\mathrm{t}})_{\mathrm{NM}}$. $\mathcal{H}_{\mathrm{MM}}$ is positive semi-definite. Thus, the Hessian matrix $\boldsymbol{H}_{\mathrm{IM}\times\mathrm{IM}}(\boldsymbol{D})$ is positive semi-definite. Consequently, we have that

J_f is a convex function with respect to $\boldsymbol{D}$ and $\boldsymbol{W}$. Consider the gradient descent nature of the learning with forgetting algorithm as shown in (6.14). We conclude the convergence of the structural learning with forgetting for minimizing (6.13).

Visual Quality. In general, (6.12) shows that the best achievable visual quality for an obtained down-sampling algorithm is limited only by the pre-selected down-sampling method in the spatial domain. As mentioned in the above, extensive studies have been conducted on spatial-domain down-sampling[4]. Therefore, it is natural to use such a method with the optimal visual quality to create a reference image. Specifically, we choose the corresponding spatial-domain method based on analysis of low-pass filtering and interpolation designs. The design of low-pass filters for down-sampling normally involves a trade-off among three factors, i.e., aliasing, low-frequency components, and ringing. A filter with a sharp transition band provides a good performance on anti-aliasing and preserving low-frequency components, yet a sharp transition band incurs ringing along intensity edges in the filtered image. A popular choice of low-pass filter with a desirable trade-off among those three factors is the Butterworth filter. Therefore, in this study we use the Butterworth filter for the low-pass filtering before down-sampling. Specifically, we choose L_{M1} and R_{1N} according to two 1D Butterworth filters with the frequency response function $|H(f)| = \sqrt{\frac{1}{1+(f/f_c)^{2L}}}$, where f_c is the cutoff frequency and L represents the order, which characterizes the transition band. For the interpolation algorithms, we choose the commonly-used cubic B-spline interpolation, which provides an output with continuous second order derivative.

Apparently, an LTDS obtained in Section 6.5.3 based on the above-selected spatial-domain method is expected to inherit the advantages of anti-aliasing, ringing avoidance, and low-frequency components preservation. Later on, we will show resulting images for subjective evaluation of the visual quality, in terms of aliasing and ringing effect.

[4]By the best achievable visual quality, we mean a subjective quality evaluation based on analysis of anti-aliasing, ringing avoidance, and low-frequency components preservation, which may reflect today's best understanding about image down-sampling.

Down-sampling Ratio. Because DCT is a block-based transform, the feasible ratio for a down-sampling method in DCT domain is limited by two parameters, i.e., the size of the image and the block size of the DCT transform. Consider an M×N image and an S×S DCT transform. All possible ratios for vertically down-sampling form a set of $\boldsymbol{r}_v = \{\frac{i}{\mathrm{M_S}}, i = 1, \cdots, \mathrm{M_S}\}$, while $\boldsymbol{r}_h = \{\frac{j}{\mathrm{N_S}}, j = 1, \cdots, \mathrm{N_S}\}$ includes all possible ratios for horizontally down-sampling, where $\mathrm{M_S} = \frac{\mathrm{M}}{\mathrm{S}}$ and $\mathrm{N_S} = \frac{\mathrm{N}}{\mathrm{S}}$ are the numbers of DCT blocks along the height and the width, respectively.

In case that the vertical scaling ratio and the horizontal scaling ratio are required to be the same [5], the set for all possible ratios is $\boldsymbol{r} = \{\frac{i}{G_{cd}}, i = 1, \cdots, G_{cd}\}$, where G_{cd} is the greatest common divisor of $\mathrm{M_S}$ and $\mathrm{N_S}$. The proposed LTDS-based method is capable of dealing with any ratio in $\boldsymbol{r}$.

The proposed method, furthermore, supports a combination of any vertical down-sampling ratio $r_h \in \boldsymbol{r}_h$ and any horizontal down-sampling ratio $r_v \in \boldsymbol{r}_v$. This provides flexibility to support a ratio $r \notin \boldsymbol{r}$ without causing noticeable visual distortion by allowing a small difference between the horizontal scaling ratio and the vertical scaling ratio. Specifically, for any ratio r, the proposed method performs the down-sampling horizontally by $r_h = \frac{\mathrm{floor}(r \cdot \mathrm{N_S})}{\mathrm{N_S}}$ and vertically by $r_v = \frac{\mathrm{floor}(r \cdot \mathrm{M_S})}{\mathrm{M_S}}$. In general, the distortion to the image proportion caused by such a small difference between r_h and r_v is virtually unnoticeable. Moreover, the flexibility of allowing the difference between r_h and r_v makes it straightforward to adapt the output image to a specific displaying resolution. For example, consider a picture of original size 480×720, a typical handset displaying resolution of 240×320, and 8×8 DCT size. The proposed method will process the image by $r_v = 2 : 1$ and $r_h = 2.25 : 1$ for a full screen display. On the other hand, a method based on DCT coefficient manipulation has to cut 80 columns in the original image or to pad 24 blank rows to the image in order to display with the full screen.

[5] It is ideal that down-sampling can be carried out with the same scaling ratio along the height and the width. However, in many cases, a little difference between the vertical scaling ratio and the horizontal scaling ratio is acceptable as long as it does not cause noticeable distortion to viewers.

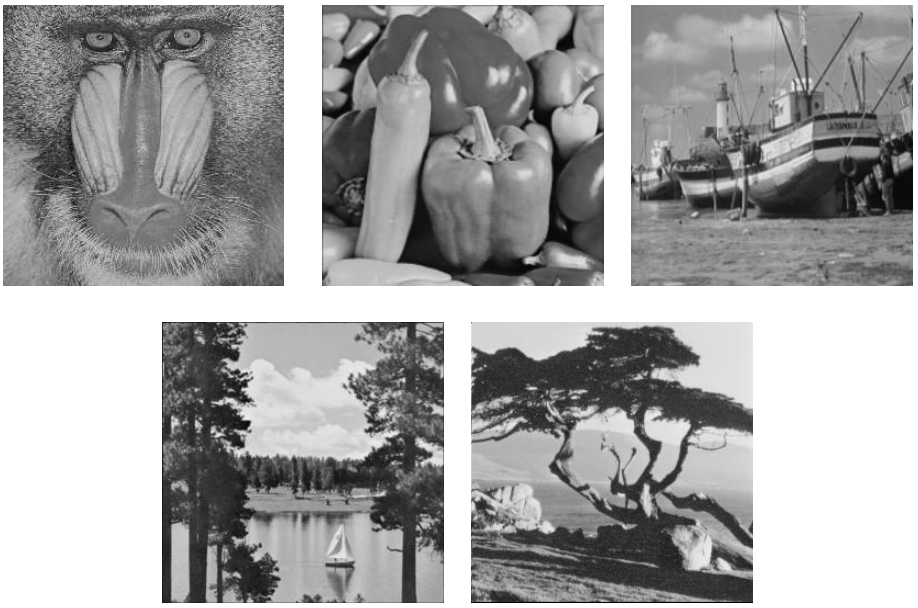


Figure 6.8: Five images used for building up the training set.

6.6 Experimental Results

The proposed design algorithm has been implemented and applied to generate a series of down-sampling methods in the DCT domain according to various user preferences to the relative significance of visual quality over the complexity. A spatial-domain method with the 10th order Butterworth low-pass filtering and cubic B-spline interpolation is selected to generate reference images for evaluating the visual quality among different LTDSs. Then, we compare the obtained LTDSs with other DCT-domain methods for down-sampling ratios being 2:1 and 3:2, respectively.

Table 6.1 shows the performance of three LTDSs for down-sampling with a ratio of 2:1 obtained using the proposed method. Experiments for finding the optimal LTDS according to user preference start with choosing 5 images $\{\boldsymbol{C}_{256\times256,i}, i = 1, \cdots, 5\}$, as shown in Figure 6.8. A reference set $\{\boldsymbol{V}_{128\times128,i}, i = 1, \cdots, 5\}$ is built up using the selected spatial-domain down-sampling method. The training for solving (6.13) begins with initializing all connections by random numbers uniformly distributed in $[-0.5, 0.5]$. The learning factor is $\alpha = 1\times10^{-6}$, $\rho = 0.5$, $\lambda = 0.1$,

Figure 6.9: Comparison of visual quality for downsampling "Lena" by 2:1 using six methods: (b) the pre-selected spatial domain method with Butterworth low pass filtering and cubic B-spline interpolation, (c) our method obtained by solving (6.12) with ($d_0 = w_0 = 0.1$), (d) the method in [12], (e) the M/L method in [58], (f) the bilinear interpolation method in [53], and (g) a fast approximate algorithm with bilinear interpolation in [53]. (a) is the original image with full resolution. Compare the visual quality of down-sampled images in (b) to (g). There are major artifacts shown in (g), e.g., at the shoulder and along the brim of the hat.

Figure 6.10: Downsampling "Barbara" by 2:1 using six methods: (b) the pre-selected spatial domain method with Butterworth low pass filtering and cubic B-spline interpolation, (c) our method obtained by solving (6.12) with ($d_0 = w_0 = 0.1$), (d) the method in [12], (e) the M/L method in [58], (f) the bilinear interpolation method in [53], and (g) a fast approximate algorithm with bilinear interpolation in [53]. (a) is the original image with full resolution. Compare the down-sampled images and pay attention to the strips on the top-left corner and the knees. Due to the lack of low-pass filtering in the bilinear interpolation method, slight aliasing is observed in the image of (f), e.g., the erroneous pattern at the left-top corner and the pepper noise at the knees. The image of (g) shows severe aliasing and artifacts.

Table 6.1: Performance of three LTDS-based methods for down-sampling DCT images by 2:1 obtained using the proposed method corresponding to three set of training parameters. The PSNR is calculated based on reference images obtained using the pre-selected down-sampling method in the spatial domain as discussed in Section 6.5.3.

Training parameters	Complexity			Visual quality
	MUL	ADD	SHL	PSNR
$d_0 = w_0 = 0.2$	0	1	1	30.4dB
$d_0 = w_0 = 0.1$	0	5.06	3.65	38.5dB
$d_0 = w_0 = 0.005$	0	17.25	13.75	46.2dB

$\eta = 0.02$. Different thresholds d_0, w_0 result in different trade-offs between distortion and complexity.

In general, the LTDS corresponding to $d_0 = w_0 = 0.1$ makes a good choice for down-sampling in the sense that it shows a better quality and a lower complexity, compared with other algorithms. For the ratio of 2:1, we compare our LTDS obtained for $d_0 = w_0 = 0.1$ with other four algorithms, which were proposed in [12], [58], and [53], respectively. The method in [12] is developed for down-sampling by a factor of 2 based on DCT coefficient manipulation while the method in [58] shares a similar spirit of DCT coefficient manipulation, except that it is extended to support more down-sampling ratios. The work in [53] is specifically targeted for down-sampling by 2:1 with 8×8 DCT, including two algorithms. The first one is essentially an LTDS based on bilinear interpolation, i.e., to compute a new sample by averaging every 2×2 block. The other one is a fast approximate algorithm for the bilinear interpolation method.

Table 6.3 shows our comparative studies for the five down-sampling algorithms with a ratio of 2:1. We measure the complexity by the number of arithmetic operations, as well as the execution time by a software implementation on our 3.4Ghz P-IV platform. Instead of the PSNR, which we use for comparing various LTDSs

Table 6.2: Image quality by PSNR for various DCT-domain methods measured against the spatial-domain reference down-sampling method.

	lena.jpg	barbara.jpg	house.jpg
LTDS ($d_0 = w_0 = 0.1$)	40.42	38.83	40.39
Method in [12]	37.53	34.81	37.66
L/M [58]	38.01	35.40	36.67
Bilinear average in [53]	38.64	33.28	38.74
Fast algorithm in [53]	28.66	23.07	30.12

obtained by different learning parameters, the visual quality here is examined by subjective criteria, such as aliasing and ringing in the table. The PSNR measurement is not used because of the lack of common reference images. The reference images obtained by the selected standard method play a fair role for evaluating various LTDSs obtained by (6.12). But they are not suitable for comparing our LTDSs with other methods because these LTDSs take a favor from those references through the optimization of (6.12). In fact, as shown in Table 6.2, the obtained LTDS ($d_0 = w_0 = 0.1$) shows a 3 to 4dB PSNR gain over other methods, yet the down-sampled images do not look that different. We examine the visual quality for down-sampling a set of 20 images, while some are included here to support the result for the visual quality in Table 6.3. As shown in Figure 6.9, the visual quality for down-sampling the image of "lena" by our method is very similar with that by other methods in [12], [58], and [53]. Moreover, Figure 6.10 shows that the lack of low-pass filtering as in the bilinear interpolation method [53] leads to aliasing for down-sampling the image of "barbara", while Figure 6.11 demonstrates ringing effects for down-sampling the image of "house" by methods in [12] and [58]. Hence, it is fair to say that our LTDS with ($d_0 = w_0 = 0.1$) shows a visual quality no worse than others in the literature, while its complexity is lower. Overall, our LTDS achieves the best trade-off between the visual quality and the computational complexity.

Since the algorithms in [53] are also LTDS, it is interesting to look into more

Table 6.3: Performance comparison of five DCT-domain methods and the reference spatial-domain method for down-sampling at a ratio of 2:1. Complexity is measured with number of operations per pixel in the original image, while computation time is reported based on our computer with 3.4Ghz P-IV CPU. The visual quality is measured by subjective criteria for a testing set of 20 images. Note that a 'yes' means that the corresponding effect shows up for some, not necessarily all, images in the whole set, while a 'no' means that the corresponding effect has not been observed for all images in the set. See Figures 6.9, 6.10 and 6.11 for the visual quality comparison. The reference spatial-domain method uses 10×10 2D low-pass filter and bicubic interpolation.

	Complexity			Visual quality			Computation time per image
	MUL	ADD	SHL	Ringing	Aliasing	Other artifacts	
Spatial reference method	36	31	0	no	no	no	112.0ms
LTDS ($d_0 = w_0 = 0.1$)	0	5.16	3.66	no	no	no	1.6ms
Method in [12]	1.25	1.25	0	yes	no	no	2.1ms
L/M [58]	3.31	8.68	2.2	yes	no	no	6.3ms
Bilinear average in [53]	3.75	5.81	0.38	yes	yes	no	2.9ms
Fast algorithm in [53]	0	2.72	0.72	yes	yes	yes	0.9ms

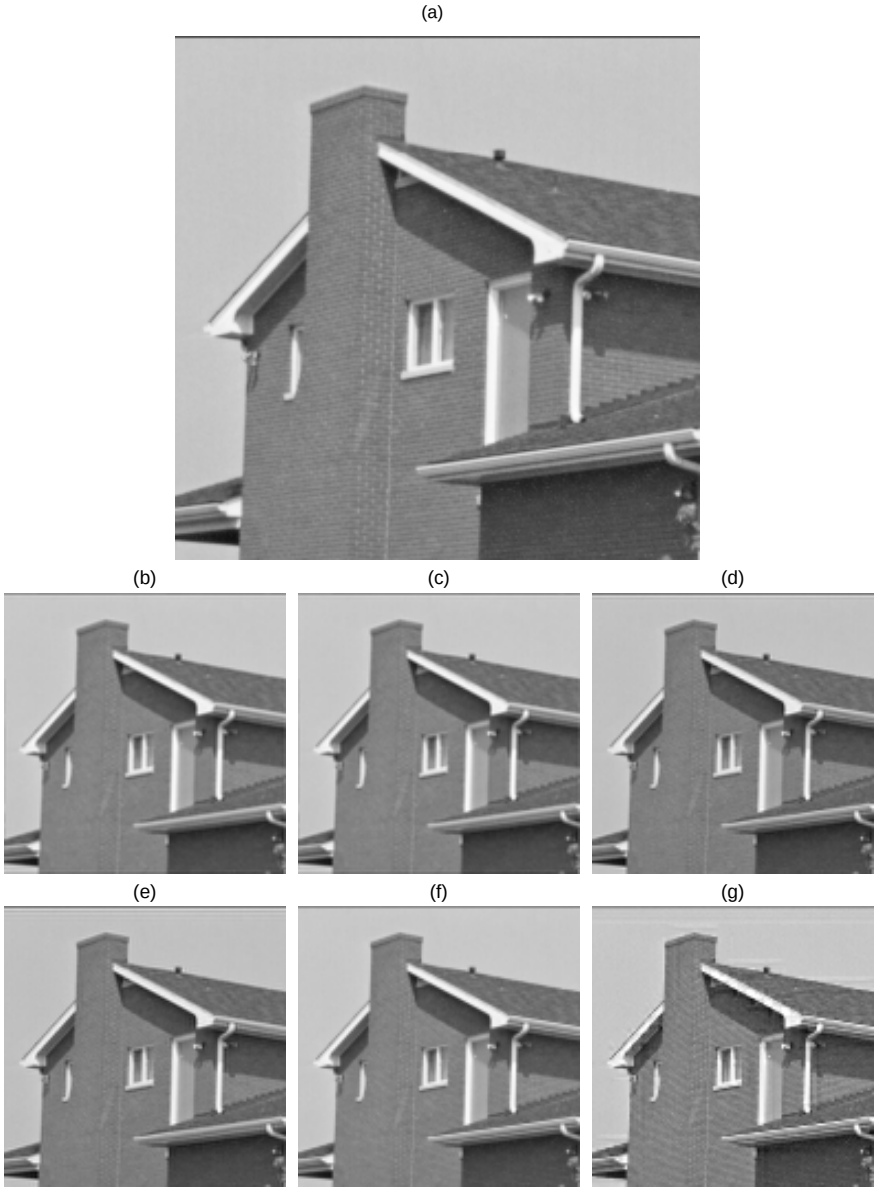


Figure 6.11: Comparison of visual quality for downsampling "House" by 2:1 using six methods: (b) the pre-selected spatial domain method with Butterworth low pass filtering and cubic B-spline interpolation, (c) our method obtained by solving (6.12), (d) the method in [12], (e) the M/L method in [58], (f) the bilinear interpolation method in [53], and (g) a fast approximate of the algorithm (f) in [53]. (a) is the original image with full resolution. Pay attention to the dark line on the top. The down-sampled images in (d) and (e) show a light line right below the dark line, indicating a typical ringing effect. Slight aliasing is observed in (f), e.g., the erroneous pattern seen along the eaves on the right. The image of (g) shows severe aliasing and artifacts.

details for the comparison between the algorithms in [53] and our obtained LTDS. The algorithms in [53] accord to a concatenation of inverse DCT, bilinear interpolation, and DCT. Essentially, the concatenation leads to an LTDS, which is shown in the following:

$$\boldsymbol{A}_1 = \begin{pmatrix} 0.500 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.453 & 0.208 & -0.037 & 0.011 & 0.000 & -0.011 & 0.037 & -0.208 \\ 0.000 & 0.500 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & -0.500 \\ -0.159 & 0.396 & 0.257 & -0.049 & 0.000 & 0.049 & -0.257 & -0.396 \\ 0.000 & 0.000 & 0.500 & 0.000 & 0.000 & 0.000 & -0.500 & 0.000 \\ 0.106 & -0.176 & 0.384 & 0.245 & 0.000 & -0.245 & -0.384 & 0.176 \\ 0.000 & 0.000 & 0.000 & 0.500 & 0.000 & -0.500 & 0.000 & 0.000 \\ -0.090 & 0.139 & -0.188 & 0.433 & 0.000 & -0.433 & 0.188 & -0.139 \end{pmatrix},$$

$$\boldsymbol{A}_2 = \begin{pmatrix} 0.500 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ -0.453 & 0.208 & 0.037 & 0.011 & 0.000 & -0.011 & -0.037 & -0.208 \\ 0.000 & -0.500 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.500 \\ 0.159 & 0.396 & -0.257 & -0.049 & 0.000 & 0.049 & 0.257 & -0.396 \\ 0.000 & 0.000 & 0.500 & 0.000 & 0.000 & 0.000 & -0.500 & 0.000 \\ -0.106 & -0.176 & -0.384 & 0.245 & 0.000 & -0.245 & 0.384 & 0.176 \\ 0.000 & 0.000 & 0.000 & -0.500 & 0.000 & 0.500 & 0.000 & 0.000 \\ 0.090 & 0.139 & 0.188 & 0.433 & 0.000 & -0.433 & -0.188 & -0.139 \end{pmatrix},$$

and the corresponding LTDS is defined as

$$\boldsymbol{D}_{\text{IM}} = \begin{pmatrix} \lfloor \boldsymbol{A}_1 \boldsymbol{A}_2 \rceil & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \ddots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \lfloor \boldsymbol{A}_1 \boldsymbol{A}_2 \rceil \end{pmatrix}$$

and

$$\boldsymbol{W}_{\text{NJ}} = \begin{pmatrix} \begin{pmatrix} \boldsymbol{A}_1^{\text{T}} \\ \boldsymbol{A}_2^{\text{T}} \end{pmatrix} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \ddots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \begin{pmatrix} \boldsymbol{A}_1^{\text{T}} \\ \boldsymbol{A}_2^{\text{T}} \end{pmatrix} \end{pmatrix}.$$

The above LTDS is further processed in [53] for fast computation. Specifically, elements in $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ are quantized to one of four levels, i.e., 0, $\frac{1}{8}$, $\frac{1}{4}$, and $\frac{1}{2}$. Consequently,

$$\boldsymbol{A}_{1,\text{fast}} = \begin{pmatrix} 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.5 & 0.25 & 0 & 0 & 0 & 0 & 0 & -0.25 \\ 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & -0.5 \\ -0.125 & 0.5 & 0.25 & 0 & 0 & 0 & -0.25 & -0.5 \\ 0 & 0 & 0.5 & 0 & 0 & 0 & -0.5 & 0 \\ 0.125 & -0.25 & 0.5 & 0.25 & 0 & -0.25 & -0.5 & 0.25 \\ 0 & 0 & 0 & 0.5 & 0 & -0.5 & 0 & 0 \\ -0.125 & 0.125 & -0.25 & 0.5 & 0 & -0.50 & 0.25 & -0.125 \end{pmatrix},$$

which is the quantization output of $\boldsymbol{A}_1$. The quantization output of $\boldsymbol{A}_2$ is similar, except the difference of the sign.

The LTDS obtained by our proposed method with $d_0 = w_0 = 0.1$, on the other hand, is as follows.

$$\boldsymbol{D}_{I\times M} = \begin{pmatrix} \lfloor \boldsymbol{B}_1 \boldsymbol{B}_2 \rfloor & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \ddots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \lfloor \boldsymbol{B}_1 \boldsymbol{B}_2 \rfloor \end{pmatrix},$$

and

$$\boldsymbol{W}_{\text{NJ}} = \begin{pmatrix} \begin{pmatrix} \boldsymbol{B}_1^{\mathrm{T}} \\ \boldsymbol{B}_2^{\mathrm{T}} \end{pmatrix} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \ddots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \begin{pmatrix} \boldsymbol{B}_1^{\mathrm{T}} \\ \boldsymbol{B}_2^{\mathrm{T}} \end{pmatrix} \end{pmatrix}.$$

where

$$\boldsymbol{B}_1 = \begin{pmatrix} 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.4375 & 0.1875 & -0.0625 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.15625 & 0.34375 & 0.25 & -0.09375 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.4375 & 0 & 0 & 0 & 0 & 0 \\ 0.0625 & -0.15625 & 0.2890625 & 0.21875 & -0.09375 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.375 & 0 & 0 & 0 & 0 \\ -0.0625 & 0.0625 & -0.125 & 0.203125 & 0.125 & -0.09375 & 0 & 0 \end{pmatrix},$$

$$\boldsymbol{B}_2 = \begin{pmatrix} 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.4375 & 0.1875 & 0.0625 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.50 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.15625 & 0.34375 & -0.25 & -0.09375 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.4375 & 0 & 0 & 0 & 0 & 0 \\ -0.0625 & -0.15625 & -0.2890625 & 0.21875 & 0.09375 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.375 & 0 & 0 & 0 & 0 \\ 0.0625 & 0.0625 & 0.125 & 0.203125 & -0.125 & -0.09375 & 0 & 0 \end{pmatrix}.$$

Consider the complexity of computing the obtained LTDS with ($d_0 = w_0 = 0.1$) for down-sampling by 2:1. The computation of the LTDS can be broken down for each 16×16 block as $\lfloor \boldsymbol{B}_1 \boldsymbol{B}_2 \rfloor \boldsymbol{C}_{16\times 16} \begin{pmatrix} \boldsymbol{B}_1^{\mathrm{T}} \\ \boldsymbol{B}_2^{\mathrm{T}} \end{pmatrix}$. Consider the binary representation of $\boldsymbol{B}_1$, i.e.,

$$\begin{pmatrix} 2^{-1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2^{-2}+2^{-3}+2^{-4} & 2^{-3}+2^{-4} & -2^{-4} & 0 & 0 & 0 & 0 & 0 \\ 0 & 2^{-1} & 0 & 0 & 0 & 0 & 0 & 0 \\ -(2^{-3}+2^{-5}) & 2^{-2}+2^{-4}+2^{-5} & 2^{-2} & -(2^{-4}+2^{-5}) & 0 & 0 & 0 & 0 \\ 0 & 0 & 2^{-2}+2^{-3}+2^{-4} & 0 & 0 & 0 & 0 & 0 \\ 2^{-4} & -(2^{-3}+2^{-5}) & 2^{-2}+2^{-5}+2^{-7} & 2^{-3}+2^{-4}+2^{-5} & -(2^{-4}+2^{-5}) & 0 & 0 & 0 \\ 0 & 0 & 0 & 2^{-2}+2^{-3} & 0 & 0 & 0 & 0 \\ -2^{-4} & 2^{-4} & -2^{-3} & 2^{-3}+2^{-4}+2^{-6} & 2^{-3} & -(2^{-4}+2^{-5}) & 0 & 0 \end{pmatrix}.$$

Denote a column vector from $\boldsymbol{C}_{16\times 16}$ as $(c_1, \cdots, c_{16})$. Consider the symmetry between $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$. The corresponding column for the left side matrix multipli-

Table 6.4: Performance of three LTDS-based methods for down-sampling DCT images by 3:2 obtained using the proposed method corresponding to three set of training parameters. The PSNR is calculated based on reference images obtained using the pre-selected down-sampling method in the spatial domain as discussed in Section 6.5.3.

	Complexity			Visual quality
	MUL	ADD	SHL	PSNR
$d_0 = w_0 = 0.1$	0.26	8.9	7.85	30.1dB
$d_0 = w_0 = 0.02$	0.26	10.4	14.3	35.6dB
$d_0 = w_0 = 0.006$	0.23	25.1	25	48.6dB

cation is computed as follows:

$2^{-1}(c_1 + c_9)$,
$(2^{-2} + 2^{-3} + 2^{-4})(c_1 + c_9) + (2^{-3} + 2^{-4})(c_2 + c_{10}) - 2^{-4}(c_3 + c_{11})$,
$2^{-1}(c_1 + c_{10})$,
$-(2^{-3} + 2^{-5})(c_1 + c_9) + (2^{-2} + 2^{-4} + 2^{-5})(c_2 + c_{10}) + 2^{-2}(c_3 + c_{11}) - (2^{-4} + 2^{-5})(c_4 + c_{12})$,
$(2^{-2} + 2^{-3} + 2^{-4})(c_3 + c_{11})$,
$2^{-4}(c_1 + c_9) - (2^{-3} + 2^{-5})(c_2 + c_{10}) + (2^{-2} + 2^{-5} + 2^{-7})(c_3 + c_{11}) + (2^{-3} + 2^{-4} + 2^{-5})(c_4 + c_{12}) - (2^{-4} + 2^{-5})(c_5 + c_{13})$,
$(2^{-2} + 2^{-3})(c_4 + c_{12})$,
$-2^{-4}(c_2 + c_{10} - c_1 - c_9) + 2^{-3}(c_5 + c_{13} - c_3 - c_{11}) + (2^{-3} + 2^{-4} + 2^{-6})(c_4 + c_{12})$.

It is easy to see that the above column consume 53 additions and 39 shifts, denoted as 53A and 39S, respectively. For each 16×16 block, the left side matrix multiplication takes 16(53A+39S), while the right side matrix multiplication requires 8(53A+39S). Therefore, the number of operations per each original pixel is $\frac{16(53A+39S)+8(53A+39S)}{256} = 5.06A + 3.65S$, which is shown in Table 6.1.

Compare the procedure of designing the fast algorithm in [53] with the proposed framework (6.12) for developing our LTDS. They share similar ideas of quantizing the coefficients. The proposed design framework, however, employs a more advanced quantization design, where the quantization is integrated into the optimization scheme, leading to a better trade-off between the computational complexity and the visual quality. In general, the learning with forgetting may be considered as quantization too, except that the criterion is to search for coefficients which contribute the least to the visual quality and quantize them to zeros.

Table 6.5: Performance comparison for down-sampling JPEG images with a ratio of 3:2.

	Complexity			Visual quality		Computation time per image
	MUL	ADD	SHL	Ringing effect	Other artifacts	
LTDS ($d_0 = w_0 = 0.02$)	0.26	10.4	14.3	no	no	3.2ms
L/M [58]	1.94	8.06	0	yes	no	4.1ms

The proposed framework is applicable for generating down-sampling algorithms with arbitrary ratios. Experiments have been conducted to generate down-sampling algorithms with a ratio of 3:2. With different training parameters, three LTDSs have been obtained for down-sampling by 3:2, as shown in Table 6.4. Note that the number of multiplications is not zero, meaning that there are some multipliers for which multiplication are not substituted with additions and shifts. This is because the binary representation as shown in (6.9) may contains too many ones. A rule of allowing at most 5 ones in the binary representation was applied in our experiments.

The obtained LTDS for down-sampling by 3:2 is compared with the L/M method in [58], since the method in [53] is for 2:1 only and the work in [12] targets for down-sampling by a factor of 2. The result is shown in Table 6.5, with a focus on the complexity. Essentially, the obtained LTDS has a lower complexity than the method in [58]. Specifically, there are two algorithms proposed for down-sampling by 3:2 in [58], referred to as case I and case II. In this comparison, the case II algorithm is used because of its lower complexity. Experimental results by the computation time show that the obtained LTDS is more efficient than the case II algorithm.

Figure 6.12 shows images down-sampled by 3:2 using three methods for two typical images, "Barbara" and "House". Mainly the interest of comparison lies on the obtained LTDS and the L/M method. For "Barbara", the resulting down-sampling images are quite similar with each other, as shown in Figure 6.12 by (b) and (c). For "House", the ringing effect is observed for the method in [58] due

to its inherent drawbacks from DCT coefficient truncation, which is equivalent to filtering with an ideal filter. Specifically, there are several slight lines on the top, indicating the occurrence of ringing. Overall, compared with the method in [58] the obtained LTDS achieves a reduced complexity and a slightly better visual quality for the down-sampling ratio of 3:2.

6.7 Chapter Summary

The goal of this chapter is to study the trade-off between distortion and complexity for images/video frames down-sampling in the DCT domain, which is motivated by image/video transcoding. Essentially, a DCT-domain down-sampling design framework has been proposed. Certainly, the proposed design framework itself does not depend on any spatial-domain method. In other words, it is open to adopt other spatial-domain methods as the reference for the visual quality, if there is any other method proven to be superior to the method selected in our experiments. As the main interest in developing DCT-domain down-sampling method is for reducing the computational complexity, we have shown that the proposed design framework yields LTDSs which are more efficient than other DCT-domain methods in the literature. It has achieved a desired trade-off between the visual quality and the computational complexity.

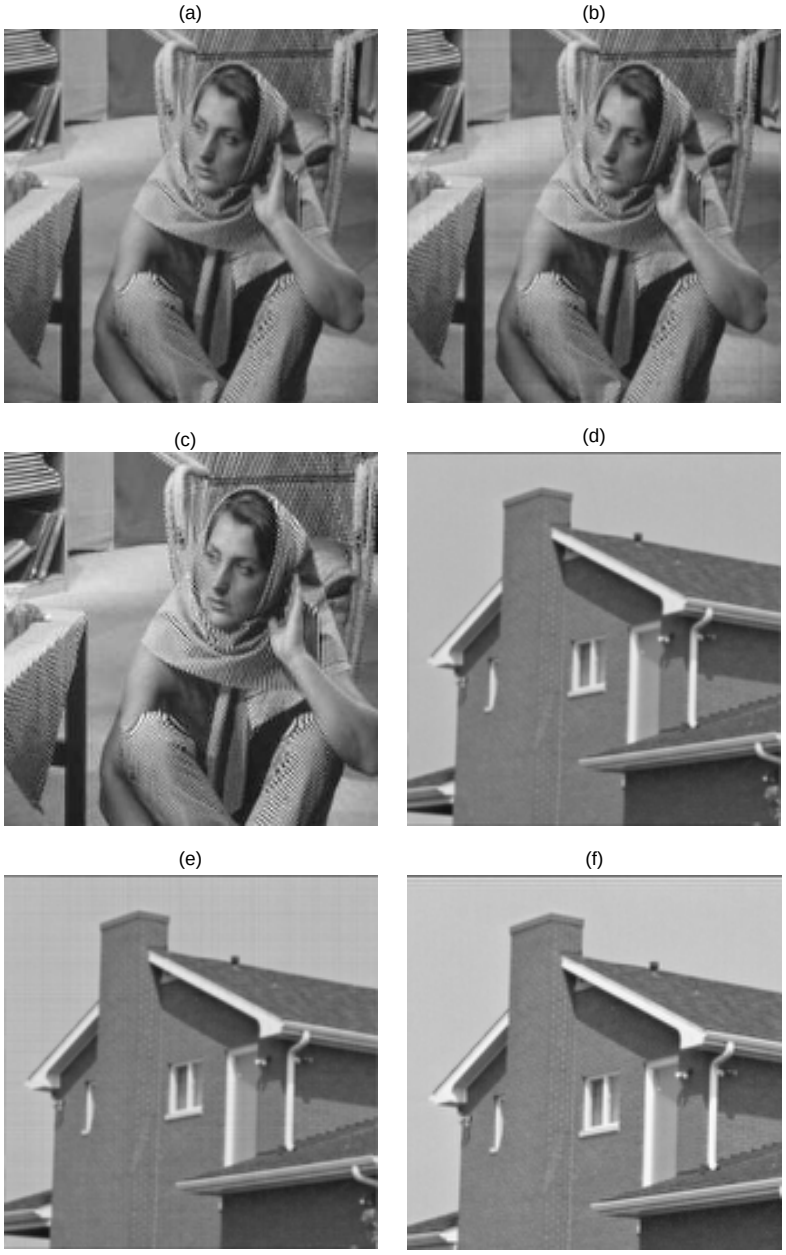


Figure 6.12: Comparison of visual quality for downsampling "Barbara" and "House" by 3:2 using three methods: (a)(d) the pre-selected spatial domain method with Butterworth low pass filtering and cubic B-spline interpolation, (b)(e) a method obtained by solving (6.12), (c)(f) the M/L method in [58]. The visual quality for the two methods are very similar, except that slight ringing is observed for the M/L method in [58], e.g., there are some light lines on the top of (f).

Chapter 7

Conclusions and Future Research

This chapter concludes the thesis with a summary of contributions and presents a few thoughts on future research.

7.1 Conclusions

In this thesis, we first study the RD optimal hybrid video coding and its application to optimize RD trade-off for H.264. Using SDQ, we have proposed a general framework in which motion estimation, quantization, and entropy coding in the hybrid coding structure for the current frame can be jointly designed to minimize the actual RD cost given previously coded reference frames. Within the framework, we have then developed three RD optimization algorithms—a graph-based algorithm for SDQ, an algorithm for residual coding optimization, and an iterative overall algorithm—with them embedded in the indicated order. Specifically, we have developed these algorithms corresponding to syntax constraints of H.264 baseline coding and H.264 main profile coding, respectively.

These algorithms have been implemented based on the reference encoder JM82 of H.264, as shown in Chapter 4 with compatibility to H.264 baseline profile and Chapter 5 with compatibility to H.264 main profile. Experiments in Chapter 4

have demonstrated that for a set of typical video testing sequences, the graph-based SDQ algorithm based on CAVLC achieves on average 6% rate reduction at the same PSNR (ranging from 30dB to 38dB) when compared with the baseline RD optimization method implemented in the H.264 reference software, and the overall optimization algorithm with baseline compatibility achieves 12% rate reduction. With a similar comparative setting, experiments in Chapter 5 have showed that the graph-based SDQ algorithm based on CABAC achieves on average 5% rate reduction over the reference main profile H.264 codec using CABAC, and the overall optimization algorithm with main profile compatibility achieves 10% rate reduction.

The main contribution in Chapter 6 is a framework for designing down-sampling method in the DCT domain by jointly optimizing the visual quality and the computational complexity. First, a linear transform model is established, based on which a joint optimization problem is formulated for finding optimal LTDSs corresponding to a pre-selected spatial-domain down-sampling method. The optimality is defined as to minimize a joint cost of the visual quality and the complexity for given parameters, which reflect the user's preference to the relative significance of the visual quality and the computational complexity. The optimization problem is addressed by modeling the LTDS with a multi-layer network and applying an automatic machine learning algorithm, i.e., structural learning with forgetting for training the network. The proposed design framework has been applied to find the optimal LTDSs corresponding to a popular spatial-domain down-sampling method with Butterworth low-pass filtering and cubic B-spline interpolation. Experiments show that the obtained LTDS inherits the desirable properties of anti-aliasing and ringing avoidance from the pre-selected spatial-domain method while being computationally efficient.

7.2 Future Research

Needless to say, there are many topics left for future work. Yet, in the following, we discuss a few of them that may be pictured in Figure 1.1 as mentioned in the introduction.

7.2.1 Hybrid Video Transcoding with Spatial Resolution Reduction

Video coding involves two major coding techniques, i.e., predictive coding and nonpredictive coding. Correspondingly, there are two types of video transcoding. Predictive video coding methods explore and utilize the cross-frame and/or cross-block similarity to achieve high compression. For example, a hybrid video coding method uses inter prediction between frames and intra prediction within frames to improve compression efficiency. All hybrid video coding standards such as MPEG-2 and H.264 employ predictive coding. On the other side, nonpredictive video coding methods process each frame in a video clip separately. They are less efficient than predictive methods in terms of compression, yet they are preferable for applications such as film and television postproduction because nonpredictive coding enables easy editing, which means that any frame in a video clip is accessible with the same ease as any other.

In this thesis, we have studied DCT-domain down-sampling, which plays a key role in transcoding nonpredictively-coded images and video clips, such as JPEG images and DV video clips. JPEG is one of the most popular formats for images on the Internet. The DV format is also widely used for consumer and professional video production. Features of the DV standard includes its standard interface of Firewire, also known as IEEE 1394, for getting video into and out of computers, and its nonpredictive compression for easy editing. Motivations for transcoding nonpredictively-coded video clips come from the fact that many video resources are originally recorded in nonpredictive formats such as DV.

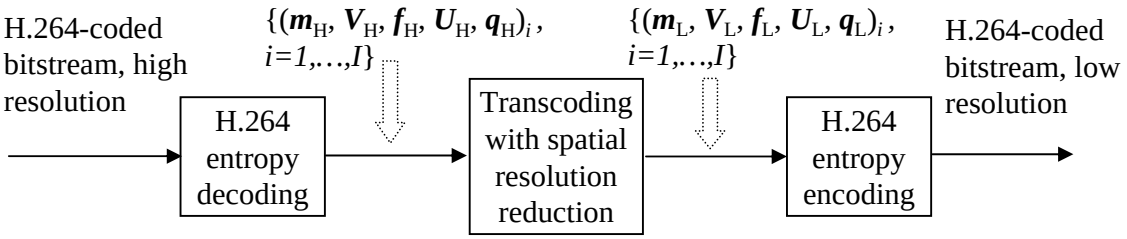


Figure 7.1: Diagram of transcoding H.264-coded video with spatial resolution reduction. The symbols of $\boldsymbol{m}$, $\boldsymbol{V}$, $\boldsymbol{f}$, $\boldsymbol{U}$, and $\boldsymbol{q}$ are prediction modes, motion vectors, reference frame numbers, quantization outputs, and quantization step sizes as defined in Chapter 3. The subscript i denotes the frame number.

Nevertheless, there are also strong motivations for transcoding predictively-coded video clips with spatial resolution reduction because of the abundance of predictively-coded video data. Since predictive video coding is also known as hybrid video coding, we call this study hybrid video transcoding. Hybrid video Transcoding involves both processing DCT coefficients and re-prediction. Particularly, a key step is to re-estimate new motion information based on old motion information obtained from the input bitstream. For example, consider to transcode an H.264-coded video clip with spatial resolution reduction, as shown in Figure 7.1. The main problem is how to find and utilize the correlation between the motion information with high spatial resolution ($\boldsymbol{m}_H$, $\boldsymbol{V}_H, \boldsymbol{f}_H$) and that with low spatial resolution ($\boldsymbol{m}_L$, $\boldsymbol{V}_L, \boldsymbol{f}_L$).

A preliminary study on transcoding H.264-coded video has been conducted in [73], where a linear method is developed to estimate a range for $\boldsymbol{V}_L$ based on $\boldsymbol{V}_H$ and then a full search within the range is performed to find ($\boldsymbol{m}_L$, $\boldsymbol{V}_L, \boldsymbol{f}_L$). Essentially, a linear relationship is assumed between motion vectors in high resolution scenes $\boldsymbol{V}_H$ and motion vectors in low resolution pictures $\boldsymbol{V}_L$. Simulations show that the linear method works fairly well to predict the range. Yet, the above transcoding method requires to fully decode the input bitstream, then to down-sample frames in the spatial domain, and to re-encoding the down-sampled frames. This procedure is slow and would be further accelerated if ($\boldsymbol{m}_L$, $\boldsymbol{V}_L, \boldsymbol{f}_L$) could be

computed directly based on ($\boldsymbol{m}_{\mathrm{H}}$, $\boldsymbol{V}_{\mathrm{H}}, \boldsymbol{f}_{\mathrm{H}}$).

Future work on hybrid video transcoding with spatial resolution reduction may be formulated as an optimization problem for minimizing a transcoding distortion over a transform from ($\boldsymbol{m}_{\mathrm{H}}$, $\boldsymbol{V}_{\mathrm{H}}, \boldsymbol{f}_{\mathrm{H}}$) to ($\boldsymbol{m}_{\mathrm{L}}$, $\boldsymbol{V}_{\mathrm{L}}, \boldsymbol{f}_{\mathrm{L}}$), which is conditioned on a given down-sampling method with a given ratio. The distortion may be defined based on mean square error between a frame reconstructed from ($\boldsymbol{m}_{\mathrm{L}}$, $\boldsymbol{V}_{\mathrm{L}}, \boldsymbol{f}_{\mathrm{L}}$) and a down-sampled version of a frame reconstructed from ($\boldsymbol{m}_{\mathrm{H}}$, $\boldsymbol{V}_{\mathrm{H}}, \boldsymbol{f}_{\mathrm{H}}$). Then, various non-linear optimization algorithms such as neural networks or genetic algorithms can be investigated to solve the problem.

7.2.2 Temporal Resolution Conversion for Video Sequences

As shown in Figure 1.1, a fundamental diversity between a video capturing device and a display device is the temporal resolution (also called frame rate). A well-known example is the difference between NTSC and PAL. NTSC is used in North America and it supports a temporal resolution of 29.97frames/second; PAL is adopted in Europe and it defines a resolution of 25frames/second. Temporal resolution conversion is needed when playing PAL or NTSC recorded video on NTSC or PAL devices.

The temporal resolution conversion issue is not new to the digital video precessing community at all. E.g., a motion-adaptive method was developed in [4] for frame rate up-conversion. However, existing methods either adopt a too simple scheme to provide satisfactory video quality or involve extensive computation. A recent conversation with a principal engineer in an internationally leading company for video products revealed that an effective and efficient temporal resolution conversion method is still on the most-wanted list. This serves as a strong motivation for developing practical temporal resolution conversion methods.

7.2.3 Video Coding with Side-Information Assisted Refinement

Lossy video compression today is going in two main directions, i.e., the conventional hybrid coding and the emerging Wyner-Ziv (W-Z) video coding (also called distributed video coding) [23]. Hybrid coding features a complex encoder and a simple decoder, where the encoder carries the computational burden to exploit source statistics for achieving efficient compression. W-Z video coding, as the dual to hybrid coding, enables fast encoding by shifting the bulk of computation to the decoder. Apparently, hybrid video coding is suitable for applications with powerful front devices. For example, in filming industries, video encoders enjoy high-end professional equipments with super computation power. On the other side, W-Z video coding is desirable for a system, which has more power at the decoder than at the encoder, e.g., a remote surveillance system with a powerful home station.

An interesting future study on video coding is to combine hybrid coding and W-Z coding. E.g., a new paradigm of video coding with side-information-assisted refinement is shown in Figure 7.2. It will allow us to flexibly distribute the computation burden between the encoder and the decoder by combining hybrid coding and W-Z coding in a scalable coding scheme. This research should yield helpful insight into the next generation video coding. While the current market-dominator, hybrid coding, experiences difficulties for many mobile applications due to the complex encoding, the emerging W-Z video coding is also limited by its high decoding complexity. This research allows a flexible distribution of complexity between encoding and decoding, which will open the door for many applications such as video messaging or video telephony with mobile terminals at both ends.

In addition, the combination of hybrid coding and W-Z coding in a scalable structure has another advantage, i.e., fast reviewing or searching for video data, which is desired for applications such as digital video library. By its nature of complex decoding, W-Z-coded video data suffer from slow reviewing/searching. To tackle this issue, a transcoding scheme has been studied by Girod et. al. as

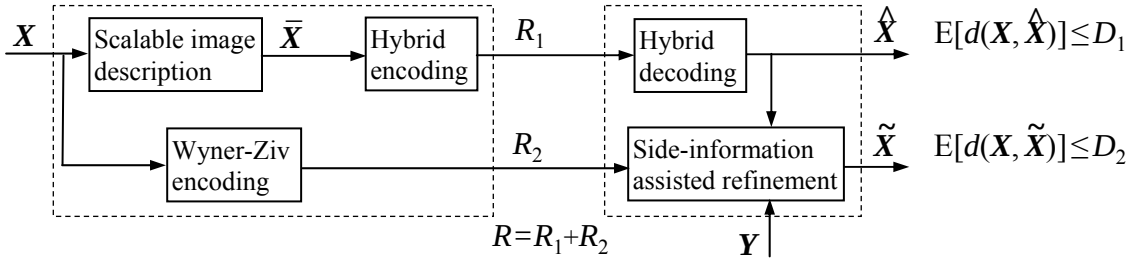


Figure 7.2: A new paradigm of video compression with side-information-assisted refinement.

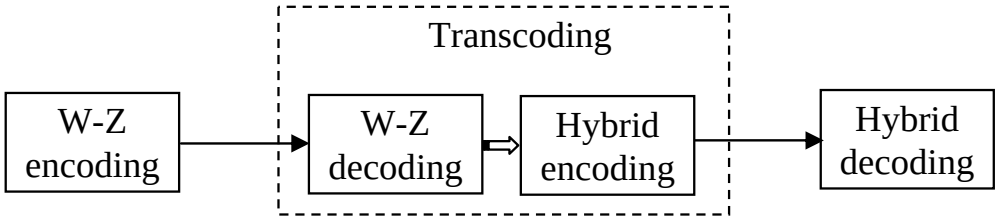


Figure 7.3: Transcoding from W-Z video coding to hybrid video compression [23].

shown in Figure 7.3, which implements a concatenation of W-Z decoding and hybrid encoding. This transcoding structure requires extra infrastructure. In the scalable combination structure of Figure 7.2, however, the fast reviewing feature is a natural result of reconstructing $\hat{X}$.

Theoretically, hybrid video coding and W-Z coding have been well studied in the literature, as the RD theory was created by Shannon in the 1950s [1] and the Slepian-Wolf theorem [70] and the W-Z theorem [80] were established in the 1970s. For the new coding scheme of video coding with side-information-assisted refinement, the following theoretic problems arise:

1. What is the RD achievable region of (R_1, R_2, D_1, D_2)?

2. What is the gap between the RD functions $R(D_2)$ with $R_1 = 0$ and the function $R(D_2)$ with given D_1?

3. What is the minimum R given two distortion levels D_1 and D_2?

Algorithmically, video compression is still challenging researchers over the world. For the new video coding diagram, the following practical designs are interesting.

1. Algorithm design for generating a base-layer description X corresponding to given R_1.

2. Algorithm design for refining $\widetilde{X}$ with $\hat{X}$ and side information Y to approach $R(D_2)$ with given D_1.

References

[1] T. Berger. *Rate Distortion Theory-A Mathematical Basis for Data Compression*. Englewood Cliffs, NJ: Prentice-Hall, 1971. 3, 14, 15, 131

[2] D. P. Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods*. Academic Press, 1982. 38

[3] G. Bjntegaard and K. Lillevold. Context-adaptive vlc (cavlc) coding of coefficients. *JVT-C028, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, 3rd Meeting*, May 2002. 29, 50, 56

[4] R. Castagno, P. Haavisto, and G. Ramponi. A method for motion adaptive frame rate up-conversion. *IEEE Transactions on Circuits and Systems for Video Technology*, 6(5):436–446, Oct. 1996. 129

[5] S.F. Chang and D. G. Messerschmitt. Manipulation and compositing of mc-dct compressed video. *IEEE Journal on Selected Areas in Communications*, 13(1):l–11, Jan. 1995. 94

[6] P. A. Chou, T. Lookabaugh, and R. M. Gray. Entropy-constrained vector quantization. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37:31–42, Jan. 1989. 35

[7] J. R. Corbera and D. L. Neuhoff. Optimal bit allocations for lossless video coders: Motion vectors vs. difference frames. *Proceedings of International Conference on Image Processing, 1995*, 3:180–183, Oct. 1995. 12

[8] J. R. Corbera and D. L. Neuhoff. Optimizing motion-vector accuracy in block-based video coding. *IEEE Transaction on Circuits and Systems for Video Technology*, 11(4):497–511, Apr. 2001. 12

[9] T. Cover and J. Thomas. *Elements of Information Theory*. 18

[10] M. Crouse and K. Ramchandran. Joint thresholding and quantizer selection for transform image coding: Entropy constrained analysis and applications to baseline jpeg. *IEEE Trans. Image Processing*, 6:285–297, Feb. 1997. 35, 41, 44

[11] W. Ding and B. Liu. Rate control of mpeg video coding and recording by rate quantization modeling. *IEEE Transactions on Circuits and Systems for Video Technology*, 6:12–20, Feb. 1996. 32

[12] R. Dugad and N. Ahuja. A fast scheme for image size change in the compressed domain. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(4):461–474, Apr. 2001. 94, 95, 99, 112, 113, 114, 115, 116, 117, 122

[13] W. Effelsberg and R. Steinmetz. *Video Compression Techniques*. Dpunkt.Verlag, 1998. 3, 20, 24

[14] T. Eude, R. Grisel, H. Cherifi, and R. Debrie. On the distribution of the dct coefficients. *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 5:365–368, Apr. 1994. 14

[15] H. Everett. Generalized lagrange multiplier method for solving problems of optimum allocation of resources. *Operations Research*, 11(3):399–417, Jun. 1963. 33

[16] M. Flierl and B. Girod. Generalized b pictures and the draft h.264/avc videoo-compression standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):587–597, Jul. 2003. 12

[17] M. Flierl, T. Wiegand, and B. Girod. Rate-constrained multihypothesis prediction for motion-compensated video compression. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(11):957–969, Nov. 2002. 12

[18] A. Gersho. On the structure of vector quantizers. *IEEE Transactions on Information Theory*, IT-28:157–166, Mar. 1982. 15, 17

[19] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression*. Boston: Kluwer Academic Publishers, 1992. 13, 15

[20] B. Girod. The efficiency of motion-compensating prediction for hybrid coding of video sequences. *IEEE Journal on Selected Areas in Communications*, SAC-5(7):1140–1154, Aug. 1987. 1, 12, 13, 14

[21] B. Girod. Motion-compensating prediction with fractional-pel accuracy. *IEEE Transactions on Communications*, 41(4):604–612, Apr. 1993. 11, 12

[22] B. Girod. Efficiency analysis of multihypothesis motion-compensated prediction for video coding. *IEEE Transactions on Image Processing*, 9(2):173–183, Feb. 2000. 12, 13, 14, 22, 27

[23] B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero. Distributed video coding. *Proceedings of the IEEE, Special Issue on Video Coding and Delivery*, 93(1):71–83, Jan. 2005. 13, 130, 131

[24] H. Gish and N. J. Pierce. Asymptotically efficient quantizing. *IEEE Transactions on Information Theory*, IT-14:676–683, Sep. 1968. 16

[25] V.K. Goyal. Transform coding with integer-to-integer transforms. *IEEE Transactions on Information Theory*, 46:465–473, Mar. 2000. 13

[26] R. M. Gray. *Source Coding Theory*. Norwell, Kluwer, 1990. 13, 14, 15, 16, 17, 18, 36

[27] E. h. Yang and J. C. Kieffer. Simple universal lossy data compression schemes derived from the lempel-ziv algorithm. *IEEE Transactions on Information Theory*, 42:239–245, Jan. 1996. 35

[28] E. h. Yang and L. Wang. Joint optimization of run-length coding, huffman coding and quantization table with complete baseline jpeg decoder compatibility. *U.S. patent application*, 2004. 18, 41, 44

[29] E. h. Yang and S. y. Shen. Distortion program-size complexity with respect to a fidelity criterion and rate distortion function. *IEEE Transactions on Information Theory*, IT-39:288–292, 1993. 40

[30] E. h. Yang and X. Yu. Soft decision quantization for h.264 with main profile compatibility. *Submitted to IEEE Transaction on Circuit and Systems for Video Technology*. Manuscript number 1602. 41

[31] E. h. Yang and X. Yu. On joint optimization of motion compensation, quantization and baseline entropy coding in h.264 with complete decoder compatibility. *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages II325–328, Mar. 2005. 39

[32] E. h. Yang and X. Yu. Optimal soft decision quantization design for h.264. *Proc. of the 9th Canadian Workshop on Information Theory*, pages 223–226, Jun. 2005. 17, 41, 43

[33] E. h. Yang and X. Yu. Rate distortion optimization of h.264 with main profile compatibility. *IEEE International Symposium on Information Theory*, pages 282–286, Jul. 2006. 41

[34] E. h. Yang and X. Yu. Rate distortion optimization for h.264 inter-frame coding: A general framework and algorithms. *IEEE Trans. On Image Processing*, 16(7):1774–1784, Jul. 2007. 43, 82

[35] E. h. Yang and J. Zeng. Method, system, and software product for color image encoding. *US application 10/831,656*, Apr. 2004. 5, 41

[36] E. h. Yang and Z. Zhang. Variable-rate trellis source encoding. *IEEE Transactions on Information Theory*, 45:586–608, Mar. 1999. 16, 17, 35, 38, 41

[37] E. h. Yang, Z. Zhang, and T. Berger. Fixed-slope universal lossy data compression. *IEEE Transactions on Information Theory*, 43(5):1465–1476, Sep. 1997. 35, 38, 40

[38] H. M. Hang and J. J. Chen. Source model for transform video coder and its application-part i: Fundamental theory. *IEEE Transactions on Circuits and Systems for Video Technology*, 7:287–298, Apr. 1997. 33

[39] H. M. Hang and J. J. Chen. Source model for transform video coder and its application-part ii: Variable frame rate coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 7:299–311, Apr. 1997. 33

[40] J. Hanson. *Understanding Video: Applications, Impact and Theory.* 3, 11

[41] Z. He and S. K. Mitra. A unified rate-distortion analysis framework for transform coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(12):1221–1236, Dec. 2001. 13

[42] HHI. H.264 reference software. *http://bs.hhi.de/ suehring/tml/.* 64

[43] M. Ishikawa. Structral learning with forgetting. *Neural Networks*, 9(3):509–521, 1996. 100, 103, 105

[44] A. K. Jain. *Fundamentals of Digital Image Processing.* Prentice Hall, NJ, 1989. 94, 99

[45] N. Kamaci and Y. Altunbasak. Frame bit allocation for h.264 using cauchy-distribution based source modelling. *Proc. of the 2005 International Conference on Acoustics, Speech, and Signal Processing*, pages II57–60, Mar. 2005. 32

[46] M. Kaneko, Y. Hatori, and A. Koike. Improvements of transform coding algorithm for motion-compensated interframe prediction errors-dct/sq coding. *IEEE Journal on Selected Areas in Communications*, 5:1068–1078, Aug. 1987. 14

[47] J. C. Kieffer. A survey of the theory of source coding with a fidelity criterion. *IEEE Transactions on Information Theory*, 39:1473–1490, Sep. 1993. 6, 37, 38

[48] J.B. Lee and A. Eleftheriadis. 2-d transform-domain resolution translation. *IEEE Transactions on Circuits and Systems for Video Technology*, 10(5):704–714, Aug. 2000. 94, 95, 98

[49] K. W. Lim, K. W. Chun, and J. B. Ra. Improvement on image transform coding by reducing interblock correlation. *IEEE Transactions on Image Processing*, 4:1146–1150, Aug. 1995. 14

[50] F. L. Luo and R. Unbehauen. *Applied Neural Networks for Signal Processing.* Cambridge University Press, 1997. 104

[51] D. Marpe, H. Schwarz, and T. Wiegand. Context-based adaptive binary arithmetic coding in the h.264/avc video compression standard. *IEEE Transaction on Circuits and Systems for Video Technology*, 13(7):620–636, Jul. 2003. 18, 19, 29, 72, 77

[52] N. Merhav and V. Bhaskaran. Fast algorithms for dct-domain image downsampling and for inverse motion compensation. *IEEE Transactions on Circuits and Systems for Video Technology*, 7(3):468–476, Jun. 1997. 4, 94, 95, 98, 99

[53] B. K. Natarajan and V. Bhaskaran. A fast approximate algorithm for scaling down digital images in the dct domain. *Proc. IEEE Int. Conf. Image Processing'1995*, pages 241–243, 1995. 99, 112, 113, 114, 115, 116, 117, 118, 119, 121, 122

[54] Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG. Adaptive quantization encoding technique using an equal expected-value rule. *ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6*, Jan. 2005. 78

[55] A. Ortega and K. Ramchandran. Rate-distortion methods for image and video compression. *IEEE Signal Processing Magazine*, pages 23–49, Nov. 1998. 1, 15, 35, 42

[56] H.W. Park, Y.S. Park, and S.K. Oh. L/m-fold image resizing in block-dct domain using symmetric convolution. *IEEE Transactions on Image Processing*, 12(9):1016–1034, Sep. 2003. 95, 99

[57] Y.S. Park and H.W. Park. Design and analysis of an image resizing filter in the block-dct domain. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(2):274–279, Feb. 2004. 99

[58] Y.S. Park and H.W. Park. Arbitrary-ratio image resizing using fast dct of composite length for dct-based transcoder. *IEEE Transactions on Image Processing*, 15(2):494–500, Feb. 2006. 94, 95, 96, 112, 113, 114, 115, 116, 117, 122, 123, 124

[59] D. E. Pearson. *Transmission and Display of Pictorial Information*. Pentech Press, London, 1975. 15

[60] W. K. Pratt. *Digital Image Processing*. John Wiley & Sons, Inc, 1991. 87, 88, 89, 93, 94, 95

[61] 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*, 3(5):533–545, Sep. 1994. 33

[62] K. Ramchandran and M. Vetterli. Rate-distortion optimal fast thresholding with complete jpeg/mpeg decoder compatibility. *IEEE Trans. Image Processing*, 3:700–704, Sep. 1994. 41, 44

[63] I. E. G. Richardson. *H.264 and MPEG-4 video compression : video coding for next generation multimedia*. Chichester ; Hoboken, NJ : Wiley, 2003. 1, 2, 11, 15, 22, 24, 29

[64] M. J. Riely and I. Richardson. *Digital Video Communications*. Artech House, Boston, 1997. 3, 21, 24

[65] D. Salomon. *Data Compression*. Springer-Verlag new York, 2004. 11

[66] K. Sayood, J. D. Gibson, and M. C. Rost. An algorithm for uniform vector quantizer design. *IEEE Transactions on Information Theory*, IT-30:805–814, Nov. 1984. 16

[67] B. Schumitsch, H. Schwarz, and T. Wiegand. Inter-frame optimization of transform coefficient selection in hybrid video coding. *Proc. of Picture Coding Symposium*, Dec. 2004. 34, 41, 44

[68] Y. Shoham and A. Gersho. Efficient bit allocation for arbitrary set of quantizers. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(9), Sep. 1988. 17

[69] H. Shu and L.-P. Chau. The realization of arbitrary downsizing video transcoding. *IEEE Transactions on Circuits and Systems for Video Technology*, 16(4):540–546, Apr. 2006. 4, 94

[70] D. Slepian and J.K. Wolf. Noiseless coding of correlated information sources. *IEEE Transactions on Information Theory*, IT-19:471–480, July 1973. 131

[71] G. J. Sullivan and T. Wiegand. Rate-distortion optimization for video compression. *IEEE Signal Processing Magazine*, pages 74–90, Nov. 1998. 33

[72] A. Tamhankar and K. R. Rao. An overview of the h.264/mpeg-4 part 10. *EC-VIP-MC 2003, 4th EURASIP Conference*, 2(5):1–51, Jul. 2003. 10

[73] J. Wang, E. h. Yang, and X. Yu. An efficient motion estimation method for h.264-based video transcoding with spatial resolution conversion. *Proceedings of IEEE International Conference on Multimedia and Expo*, pages 444–447, Sept. 2007. 128

[74] J. Wen, M. Luttrell, and J. Villasenor. Trellis-based r-d optimal quantization in h.263+. *IEEE Transaction on Image Processing*, 9(8):1431–1434, Aug. 2000. 34, 41, 44

[75] T. Wiegand and B. Girod. Lagrangian multiplier selection in hybrid video coder control. *Proceedings of ICIP'2001*, pages 542–545, Oct. 2001. 33, 46

[76] T. Wiegand, M. Lightstone, D. Mukherjee, T.G. Campbell, and S.K. Mitra. Rate-distortion optimized mode selection for very low bit rate video coding and the emerging h.263 standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 6(2):182–190, Apr. 1996. 34

[77] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. J. Sullivan. Rate-constrained coder control and comparison of video coding standards. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):688–703, Jul. 2003. 4, 13, 26, 33, 34, 35, 48, 49, 64, 66, 67, 69, 71, 81, 82, 85, 86

[78] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra. Overview of the h.264/avc video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7), Jul. 2003. 10, 25

[79] T. Wiegand, G. J. Sullivan, and A. Luthra. Draft itu-t rec. h.264/iso/iec 14496-10 avc. *Joint Video Team of ISO/IEC MPEG and ITU-T VCEG*, 2003. 4, 39, 40, 45

[80] A. Wyner and J. Ziv. The rate-distortion function for source coding with side information at the receiver. *IEEE Transactions on Information Theory*, IT-22:1–11, 1976. 131

[81] A. xViterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, April 1967. 63

[82] X. Yu, E. h. Yang, and H. Wang. Down-sampling design in dct domain with arbitrary ratio for image/video transcoding. *Submitted to IEEE Transaction on Image Processing*. Manuscript number: TIP-03195-2007. 87

[83] J. Ziv. On universal quantization. *IEEE Transactions on Information Theory*, IT-31:344–347, May 1985. 15

[84] J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, IT-23:337–342, May 1977. 5, 41

[85] J. Ziv and A. Lempel. Compression of individual sequences via variable-rate coding. *IEEE Transactions on Information Theory*, IT-24(5):530–536, Sep. 1978. 5, 41