Optimized techniques for video compression and enhancement

Nguyen, Viet Anh

2009

Nguyen, V. A. (2009). Optimized techniques for video compression and enhancement.
Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/19311>

<https://doi.org/10.32657/10356/19311>

Optimized Techniques for Video Compression and Enhancement

Nguyen Viet Anh

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement for the degree of
Doctor of Philosophy

2009

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research done by me and has not been submitted for a higher degree to any other University or Institute.

.....

Date

.....

Nguyen Viet Anh

*To my parents,
To my beloved daughter and wife,
for their love and support.*

Acknowledgements

First of all, I would like to express my deep and sincere gratitude to my supervisor, Associate Prof. Tan Yap Peng, for his invaluable guidance and strong support throughout the PhD program. It was my great honor to have him as my supervisor since my undergraduate study. He provided me with not only the in-depth knowledge, but also strong encouragement during the toughest times of the research and inspired me with the spirit of exploring science. This work would not have been possible without him.

Secondly, I would like to acknowledge Singapore Millennium Foundation (SMF) and School of EEE, NTU for awarding me the research scholarship and providing with the excellent research facilities and resources.

Thirdly, I want to thank other members of our research group for their help and cooperation. Special thanks go to Dr. Chen Zhenzhong for our many discussions and valuable advices, which certainly helped me staying at the right track.

Lastly and most importantly, I am indebted to my entire family, my wife Nguyen Anh Thu, my little daughter, my parents, parents-in-law, and my sister for their loving supports. They spent great efforts and sacrificed a lot in order for me to concentrate on my research and complete my chosen path. To them I dedicate this thesis.

Contents

Acknowledgements	i
Summary	vi
List of Figures	ix
List of Tables	xvii
1 Introduction	1
1.1 Background and Motivations	1
1.2 Thesis Scope and Main Contributions	7
1.3 Thesis Organization	10
1.4 Summary	11
2 Background and Literature Review	12
2.1 Video Compression Fundamentals	13
2.1.1 Preliminaries	13
2.1.1.1 Digital Video	13
2.1.1.2 Video Frame Format	16

2.1.1.3	Video Quality Measure	18
2.1.2	Principles of Video Compression	19
2.1.2.1	Block-Based Motion Estimation	22
2.1.2.2	Transform Coding	26
2.1.2.3	Quantization	27
2.1.2.4	Entropy Coding	29
2.1.3	Representations of Compressed Video	30
2.1.3.1	Layer of Video Data	30
2.1.3.2	Picture Type	32
2.2	Video Coding Standards	35
2.2.1	H.263 and MPEG-2	36
2.2.2	H.264/AVC	38
2.3	Related Work	41
2.3.1	Block-Matching Motion Estimation	41
2.3.2	Video Transcoding	48
2.3.3	Quality Enhancement by Post-processing	54
2.4	Summary	59
3	Motion Estimation Based on Integral Frame Attributes	60
3.1	Introduction	61
3.2	Integral Frame Attributes	64
3.3	Proposed Block-Matching Measures	67
3.3.1	SAD of Block Sums (SAD-BS)	68

3.3.2	SAD of Block Variances (SAD-VR)	69
3.3.3	SAD of Block Sums and Variances (SAD-SV)	70
3.3.4	Computational Complexity	71
3.4	Extended Use of Proposed BM Measures	81
3.4.1	Two-Step Approach	81
3.4.2	Partial Summation Elimination	86
3.5	Experimental Results	90
3.5.1	Proposed BM Measures	90
3.5.2	Extended Use of Proposed BM Measures	95
3.6	Summary	101
4	Video Transcoding Between H.263 and H.264/AVC Standards	104
4.1	Introduction	105
4.2	Complexity and Performance Issues in H.264/AVC Transcoding	108
4.2.1	Coding Features	108
4.2.2	H.264/AVC Rate Control	109
4.2.3	Efficient Options and Modes for Transcoding	112
4.3	Proposed H.263 to H.264/AVC Transcoding Methods	114
4.3.1	Fast Intra-Prediction Mode Selection	115
4.3.2	Motion Vector Re-estimation and Inter-Mode Selection .	118
4.3.3	Enhanced Rate Control for H.264/AVC Transcoding . .	123
4.3.3.1	Rate-Quantization Ratio Model	123
4.3.3.2	Proposed Rate Control Method	128

4.4	Experimental Results	130
4.5	Summary	142
5	Blind Video Enhancement from Multiple Compressed Copies	143
5.1	Introduction	144
5.2	Brief Overview of H.264/AVC Transform and Quantization . . .	147
5.3	Problem Formulation and Proposed Method	150
5.4	Video Alignment	159
5.5	Analytical Justification	164
5.6	Experimental Results	169
5.7	Summary	183
6	Conclusion and Future Work	189
6.1	Conclusion	189
6.2	Future Work	196
	Author's Publications	200
	Bibliography	202

Summary

Today, with the rapid development of cost-effective and popular digital recording devices, digital video contents are being generated and distributed at a phenomenal rate. As the volume of digital video is notoriously huge, efficient techniques to process this large amount of data are also becoming increasingly important.

The requirements for efficient video transmission and storage make video compression very important in video processing to reduce the data rate. Meanwhile, to support universal multimedia access over heterogeneous networks and devices, video transcoding that processes an existing compressed video to better suit new application constraints has also attracted much attention. To meet the growing demand for better and faster visual communication services, optimized techniques for these processes have been explored in order to achieve better performance in terms of complexity, quality, and bit rate. This requirement in turn becomes challenging due to the trade-off among these measures. Owing to the scarce resources and constraints encountered in real-time applications, video encoder and transcoder may need to reduce the complexity at the expense of degraded visual quality. When the resources and constraints are no longer concerned, how to obtain better quality from these encoded videos becomes important. To address this problem, quality enhancement by post-processing has been employed and become an active research topic in recent years. In this thesis, we propose new and efficient techniques to address key problems in video

compression and enhancement.

We first focus on optimizing the video coding process, which aims to reduce the computational complexity while maintaining an acceptable visual quality, in order to make it suitable for real-time applications. Motion estimation is mostly considered in our solution due to it being the most computationally intensive part of a typical video encoder. New block-matching measures are proposed to reduce the computational complexity of block-based motion estimation. These proposed measures evaluate the match between two blocks by making use of the features such as block sum and block variance, which can be easily computed from integral frame attributes with optimal number of computations.

Next, a complete picture of efficient format transcoding and downsizing transcoding between the existing compression standard H.263 and the state-of-the-art H.264/AVC standard is presented. To speed up the process, fast motion vector re-estimation and intra-prediction mode selection are proposed. Furthermore, an enhanced rate control is employed for H.264/AVC transcoding to improve the transcoded video quality. In particular, a model is developed to approximate the relationship between the total number of coding bits and quantization step sizes of the precoded and transcoded videos, which is used for selecting the quantization parameters at the sequence, group-of-picture, and frame levels. Also proposed is a new frame layer bit allocation scheme to achieve more accurate bit rate and constant visual quality.

Lastly, we consider the improved post-processing techniques for quality enhancement of compressed videos. We address a new research problem by blindly enhancing the quality of the video reconstructed from multiple compressed copies of the same video content. As the original source video is not always available, how to choose or derive a video of the best quality among these copies becomes challenging and interesting. Specifically, a method for reconstructing each coefficient of the video in transform domain is proposed by using a nar-

row quantization constraint set derived from the multiple compressed copies together with the Laplacian and Cauchy distribution models for each AC transform coefficient to minimize the distortion. Analytical and experiment results show that we can reconstruct a video that achieves better quality than any of the available copies.

List of Figures

1.1	Multimedia applications and communications over heterogenous networks and devices.	4
2.1	Representation of a 3-D scene over a period of time.	13
2.2	Conceptual illustration of the progressive sampling and interlaced sampling of a CIF video frame.	14
2.3	Popular pattern representations of a color video frame in the Y:Cb:Cr color space.	16
2.4	BT.601 standard formats for NTSC and PAL/SECAM TV systems.	17
2.5	Two basic stages of the video coding process.	20
2.6	Motion estimation and compensation block diagram.	21
2.7	The block diagram of a generic model of the video encoder and decoder.	22
2.8	Block-matching motion search.	23
2.9	Fractional pixel motion search.	26
2.10	Basis images of 8×8 DCT transform.	27

2.11 Illustration of a uniform scalar quantization with and without dead zone.	28
2.12 Zigzag scan order for a progressive video frame.	30
2.13 The structure of the coding unit.	31
2.14 Forward motion compensation for P-type pictures.	33
2.15 Bidirectional motion compensation for B-type pictures.	34
2.16 Example of a coded video sequence: (a) MPEG-2 (b) H.261. . .	35
2.17 Block diagrams of encoder and decoder for MPEG-2 and H.263. .	37
2.18 Block diagram of a H.264/AVC encoder.	38
2.19 4×4 luminance prediction modes.	39
2.20 Intra 16×16 prediction modes.	39
2.21 Variable block sizes for H.264/AVC motion estimation.	40
2.22 Three-step search algorithm.	43
2.23 Architecture of the fully cascaded decoder and encoder transcoder.	48
2.24 Architecture of an open-loop transcoder.	49
2.25 Architecture of the cascaded pixel-domain transcoder.	50
2.26 Cascaded frequency-domain transcoding architecture.	51
3.1 The value of integral frame attribute at pixel (m, n) is equal to the sum of some frame features over the region that is above and to the left of pixel (m, n) , inclusive, in the original frame.	65
3.2 The sum of all frame features in block D can be computed by using the four corresponding integral frame attributes at the block boundaries.	66

3.3	Example block patterns: a) 1-block, (b) 2-block, (c) 4-strip, (d) 4-block, (e) 8-strip, (f) 8-block, (g) 16-strip, (h) 16-block, and (i) SAD.	68
3.4	Motion vector of a current block can be obtained by matching the proposed BM measure with that of each candidate block within a search window in the reference frame.	69
3.5	Performance comparison for the Foreman sequence obtained by using the proposed BM measures computed with different block patterns shown in Fig. 3.3 in conjunction with the FS algorithm.	71
3.6	It takes only $(2P + 1) \times Q$ arithmetic operations to compute the proposed SAD-BS measure with a K -block pattern.	72
3.7	The gain obtained by different proposed BM measures in comparison with the FS algorithm using the conventional SAD measure.	77
3.8	Empirical hit ratios for different values of K in two cases: 1) Step1: 1-block pattern, Step 2: 16-block pattern, 2) Step 1: 2-block pattern, Step 2: 16-block pattern.	84
3.9	Flowchart of the proposed PSE scheme.	87
3.10	The number of sub-blocks used for computing PSAD-BF both with and without the prediction of the initial search center for each candidate block in the search window.	89
3.11	Performance comparison obtained by using the proposed BM measures computed with different block patterns shown in Fig. 3.3 for the Coastguard sequence.	91
3.12	Rate-distortion curves obtained by the proposed SAD-BS measure together with the preferred 4-block pattern and the conventional SAD measure for the Foreman sequence.	96

3.13	Rate-distortion curves obtained by the proposed SAD-BS measure together with the preferred 4-block pattern and the conventional SAD measure for the Tempete sequence.	97
3.14	Frame-to-frame PSNR results of the Foreman sequence obtained by the proposed SAD-BS measure and the conventional SAD measure using the FS and TSS algorithms.	98
3.15	Sample frames from the Foreman sequence encoded by using the proposed SAD-BS measure and the existing SAD measure using the FS and TSS algorithms.	99
3.16	The PSNR results obtained by the proposed two-step approach and the SAD-BS measure using the 1-block and 16-block patterns in Step 1 and Step 2, respectively, for various number of good candidate blocks retaining in Step 1.	100
3.17	The PSNR results obtained by the proposed two-step approach and the SAD-BS measure using the 2-block and 16-block patterns in Step 1 and Step 2, respectively, for various number of good candidate blocks retaining in Step 1.	100
4.1	Block diagram of the proposed video transcoder.	114
4.2	Pictorial representation of some 4×4 integer-transform coefficients of the difference between the prediction and the block to be coded in the DC prediction mode.	116
4.3	Directions of nine possible intra-prediction modes for a 4×4 block.	117
4.4	Distribution of the MVs obtained by exhaustive search around the precoded MV or the predicted MV from the neighboring macroblocks.	121

4.5	Relation between the number of coding bits in the precoded and transcoded videos by transcoding with various fixed QPs used in the H.264/AVC transcoder for the Foreman sequence.	124
4.6	Relation between the number of coding bits in the precoded and transcoded videos by transcoding with various fixed QPs used in the H.264/AVC transcoder for the Silent sequence.	125
4.7	Relation between the average ratio of total number of coding bits and the ratio of quantization step sizes in precoded and transcoded videos.	127
4.8	Frame-to-frame PSNR results of the News sequence obtained by the proposed H.264/AVC transcoding methods and the H.264/AVC recoding (RC) method.	135
4.9	Frame-to-frame PSNR results of the Foreman sequence obtained by the proposed H.264/AVC transcoding methods and the H.264/AVC recoding (RC) method.	135
4.10	Sample frames from the News sequence transcoded from the precoded H.263 video by the proposed syntax transcoding methods with quarter-pixel refinement and the cascaded H.264/AVC recoding (RC) method.	136
4.11	Sample frames from the Foreman sequence transcoded from the precoded H.263 video by the proposed syntax transcoding methods with quarter-pixel refinement and the cascaded H.264/AVC recoding (RC) method.	137
4.12	PSNR results (in dB) and the differences between the target and achieved bit rates obtained by transcoding the Foreman sequence using various starting QPs at a given target bit rate.	138

4.13	PSNR performance (in dB) of the transcoded videos of the Foreman sequence obtained by the H.264/AVC rate control method and the proposed enhanced rate control.	141
4.14	PSNR performance (in dB) of the transcoded videos of the Silent sequence obtained by the H.264/AVC rate control method and the proposed enhanced rate control.	141
5.1	The distance from frame 30 and 50 of one compressed copy to all frames of the other copy of a short video content extracted from a popular situation comedy.	161
5.2	The optimal path for frame matching between two sequences obtained by using dynamic programming.	162
5.3	The illustration of how narrow the intersection of the multiple QCSs depends not only on the relation among the sizes of QCSs from multiple compressed copies, but also the position of the QCSs' intervals.	167
5.4	Probability of the sizes of the QCS S_1 and the narrow QCS S obtained by the proposed method for different values of Q_2 . . .	168
5.5	PSNR results (dB) of the best input copy from the two available compressed copies of the Foreman and Stefan sequences and the reconstructed video obtained by using the proposed method in conjunction with the Laplacian and Cauchy models to approximate the AC coefficient distribution.	171
5.6	Sample frames of the 63th frame from the Stefan sequence obtained by the proposed method and by reconstructing from three input copies.	176

5.7	Sample frames of the 77th frame from the Stefan sequence obtained by the proposed method and by reconstructing from three input copies.	177
5.8	PSNR gain (dB) of the reconstructed video obtained by the proposed method compared with the best input copy with various target bit rates and number of copies for the Foreman and Silent sequences.	179
5.9	PSNR results (dB) of the two copies of the Tennis and Silent sequences compressed as variable and constant bit rates and the reconstructed video obtained by using the proposed method. . .	181
5.10	Real video test sequences (number of frames in parenthesis). . .	182
5.11	Sample frames of the 20th frame from the Soldier sequence obtained by the proposed method and by reconstructing from three input copies.	184
5.12	Enlarged regions of the sample frames of the 20th frame from the Soldier sequence obtained by the proposed method and by reconstructing from three input copies for better visualization of the perceptual differences.	185
5.13	Sample frames of the 64th frame from the Soldier sequence obtained by the proposed method and by reconstructing from three input copies.	186
5.14	Enlarged regions of the sample frames of the 64th frame from the Soldier sequence obtained by the proposed method and by reconstructing from three input copies for better visualization of the perceptual differences.	187

6.1 Potential application of interest for multimedia applications and services.	195
---	-----

List of Tables

2.1	Parameters of a number of intermediate digital video formats and their typical applications.	18
3.1	Numbers of arithmetic operations required to compute the proposed BM measures at each search location using different block patterns in conjunction with the FS algorithm or a fast search algorithm.	75
3.2	Numbers of arithmetic operations required by the proposed BM measures and the conventional SAD measure when used together with the FS and TSS algorithms, respectively.	76
3.3	Numbers of arithmetic operations required by the proposed BM measures using different block patterns and the complexity comparison against the FS and TSS algorithms using the conventional SAD measure for a case study with $W = 15$, $M = 352$, and $N = 288$	78

3.4	The numbers of operations required by using the fast computation scheme (FCS) of the block sum and the proposed scheme without pre-computing all sub-block sums for evaluating the SAD-BS measure in conjunction with the TSS, the gradient descent search (GDS), and motion vector field adaptive search technique (MVFAST).	80
3.5	Numbers of arithmetic operations required by the proposed BM measures in the proposed two-step approach and by the conventional SAD measure when used together with the FS algorithm.	85
3.6	The PSNR results (in dB) of ten test sequences obtained by using the proposed SAD-BS measure with various block patterns and the conventional SAD measure using the FS algorithm. (The average PSNR difference is computed against that of the SAD measure.)	92
3.7	The PSNR results (in dB) of ten test sequences obtained by using the proposed SAD-SV measure with various block patterns and the conventional SAD measure using the FS algorithm. (The average PSNR difference is computed against that of the SAD measure.)	93
3.8	The PSNR results (in dB) of ten test sequences obtained by using the proposed SAD-BS measure with various block patterns and the conventional SAD measure using the TSS algorithm. (The average PSNR difference is computed against that of the SAD measure.)	94

3.9	Average number of operations per block taken by the proposed BMA with and without using the PSE scheme for the Foreman and Stefan sequences. PSE ₂ and PSE ₁ denote the use of the PSE scheme with and without the prediction of initial search center.	102
4.1	PSNR results (in dB) obtained by the cascaded H.264/AVC recoding approach using four schemes with different combinations of MCP modes and reference frames.	113
4.2	Average and cumulative percentages of the optimal MV distribution measured at different absolute distances from the new search center in eight test sequences.	120
4.3	PSNR results and the total encoding times obtained by transcoding H.263 sequences using the cascaded H.264/AVC recoding (RC) method, the MV re-estimation method proposed in Section 4.3.2, with or without quarter-pixel refinement (refn.). . . .	132
4.4	PSNR results and the total encoding times obtained by transcoding H.263 sequences using the cascaded H.264/AVC recoding (RC) method, and the proposed H.264/AVC transcoding methods in Sections 4.3.1 and 4.3.2, with or without quarter-pixel refinement (refn.).	134
4.5	PSNR results (in dB), standard deviations of the PSNR results (σ), and achieved bit rates for the Foreman sequence using different starting QPs at a given target bit rate.	139

4.6	PSNR results (in dB), standard deviations of the PSNR results (σ), and actual bit rates obtained by the H.264/AVC recoding (RC) method, the proposed fast transcoding methods in conjunction with the existing and the proposed H.264/AVC rate control methods.	140
5.1	Quantization step size conversion function	149
5.2	Standard test video sequences	169
5.3	Coding parameters of the standard test video sequences	172
5.4	Average PSNR results (in dB) of the best input copy and the video reconstructed by the proposed method from multiple input copies compressed with the same GOP structure at different target bit rates.	173
5.5	Average PSNR results (in dB) of the best input copy and the video reconstructed by the proposed method from multiple input copies compressed with different GOP structures at different target bit rates.	174
5.6	Coding parameters of the real video sequences	182
5.7	Average PSNR results (in dB) of the best input copy and the video reconstructed by the proposed method from multiple input copies compressed for real video sequences with different GOP structures at different target bit rates.	183

Chapter 1

Introduction

In this chapter, we describe some background and motivations of why optimized techniques for video compression and enhancement have become increasingly important in digital video processing and communications. We also provide the thesis scope and summarize our main contributions. The organization of the thesis is then given at the end of this chapter.

1.1 Background and Motivations

In the past, video is widely known in the traditional analog form (e.g., recorded in the form of analog electrical signal, transmitted using analog amplitude modulation, and stored on magnetic tape). Due to its nature, analog video provides a very limited amount of manipulation and interactivity. Over the last couple of decades, a digital revolution has taken place and gradually replaced analog technology with new, efficient, and high quality digital video signals.

With the rapid advances and development of cost-effective and popular dig-

ital video hardware, digital video contents are being generated and distributed at a phenomenal rate. As a result, demands for efficient techniques to process this large amount of data are also becoming increasingly critical, which in turn have attracted a lot of attention in the research of video signal processing.

One of the main challenges in digital video processing is the notoriously huge volume of raw video signal. Consider a digitized NTSC video signal, which has a resolution of 720×480 pixels for the luminance component and 360×240 pixels for the two color difference components (chromina) at the frame rate of 30 frames per second (fps) and 8 bits per pixel (bpp), the raw bit rate is about $(720 \times 480 \times 30 + 360 \times 240 \times 30 \times 2) \times 8 \simeq 124$ Mbits per second. For the High-Definition Television (HDTV) picture with the spatial resolution of 1920×1080 pixels, stored as 8-bit pixels in three color components at a 60 Hz interlaced scan, the raw bit rate is approximately 1.5 Gbits per second. Although network bandwidth continues to increase dramatically and today's storage capacity of hard disk, flash memories, and optical media is greater than ever before, it is still difficult or impossible to record and transmit the raw video signal directly. For example, even the advanced optical media single-layer blue-ray disc (BD) with the storage of 25 Gigabytes (GB) can only hold about two minutes of raw HDTV data. Current Internet throughput rates, such as the popular WiMax using 802.16e standard supporting maximum data rate at 70 Mbits per second, are insufficient to handle raw video signal in real time.

To meet the requirements of efficient video transmission and storage, video compression is necessary and has played an important role in the advance of digital signal applications. The goal of video compression is not only to reduce the data rate efficiently, but also to ensure good video quality. Complexity,

quality, and bit rate are factors that measure the success of a video compression scheme. In essence, the principle of digital video compression is to reduce the statistical redundancy of the digital video signal in both spatial and temporal domains. Motion estimation and compensation is commonly used in typical video compression systems to exploit the correlation between neighboring video frames to remove the temporal redundancy [1]. Meanwhile, transform-based coding, such as discrete cosine transform (DCT) and discrete wavelet transform (DWT) [2], is widely used to exploit the local spatial correlation property of the video signal. In a lossy video compression system, quantization [3] followed by entropy coding methods such as Huffman coding [4] or arithmetic coding [5] is generally adopted to reduce the data rate of the video signal.

With the advent of efficient compression techniques, many video coding standards have been developed and standardized by the International Telecommunication Union (ITU) and ISO/IEC Moving Picture Experts Group (MPEG) such as MPEG-1/2/4 [6–8], H.261/H.263 [9, 10], and H.264/AVC [11]. To support universal multimedia access, these standards only specify the syntax of the encoder's output while defining the decoding method. However, due to the emergence of many video coding standards, more and more videos are generated and stored in various compressed forms. These different content representation formats together with the increase of the number of different networks and types of devices make the interoperability between different systems and networks become more challenging.

We present a figure (Fig. 1.1) to illustrate the scenario of diversity and heterogeneity in the area of multimedia applications and communications. To suit different characteristics of the emergence of the new classes of pervasive com-

Figure 1.1: Multimedia applications and communications over heterogeneous networks and devices.

puting devices (e.g., personal digital assistants (PDA), third generation (3G) mobile phones, and wearable computers) and the diversity of network connections, video content requires to be compressed and customized to meet the constraints of its target application such as the available transmission bandwidth, the desired spatial or temporal resolution, or the amount of decoder's buffer available at the receiving device. Consequently, videos compressed for one application may not be well suited for other applications subject to a set of more restricted constraints, e.g., a lower transmission bandwidth or a smaller display screen.

To support video communications over heterogeneous networks, a number of scalable video coding schemes have been proposed to generate compressed video bit streams that can be easily truncated to meet different application and network constraints [12–14]. The principle of scalable video coding is to compress the video into one base layer and one or several enhancement layers so that low

bit rates, spatial or temporal resolutions could be obtained by simply truncating certain layers from the original bit stream. One of the key problems is that the scalable video coding does not support interoperability between different compressed formats and usually suffers from inferior coding efficiency compared with single-layer video coding solutions. In addition, in cases where the available transmission bandwidth is not even enough to transmit the base-layer bit stream for real-time transmission or the receiving devices are not capable of processing multi-layer bit streams, the techniques are no longer applicable. Thus, there is a need for alternative solutions to alleviate such inadequacy in these applications. One of the flexible solution that has received much attention is video transcoding.

Video transcoding is a process for converting an existing compressed video from one format to different format for adaptation of channel condition and/or device's constraints [15–17]. The existing compressed video is referred to as the precoded video while the processed video is referred to as the transcoded video. Transcoding is generally used to reduce the bit rate, adjust the spatial or temporal resolution, or change the syntax of a compressed video. For example, syntax conversion is required for a precoded video to be processed by a decoder compliant to a different compression standard, such as the conversion between MPEG-4 and H.264 standards [15]. In another scenario, when the available transmission bandwidth suddenly drops and is not adequate to deliver the precoded video, the video bit rate needs to be reduced to allow real-time transmission [16]. In addition, in order for the users of mobile multimedia-capable devices with a small display screen or limited processing power can access the video originally captured in a high spatial or temporal resolution, the precoded video needs to be scaled down before being sent to such devices [17].

With the growing demand for better and faster visual communication services, the needs for the efficiency of the video coding and transcoding processes become higher and higher. Therefore, optimized techniques for these processes have been explored and attracted much research interest in video signal processing. The main goal of research activities in these areas is to optimize video encoder or transcoder to achieve better performance in terms of complexity, quality, and bit rate. This requirement is technically challenging due to the trade-off among these measures.

One of the key issues in a conventional video encoder or transcoder is the huge requirement of processing time and resources to obtain high coding efficiency, especially the motion estimation process that exhibits a high computational burden of typically 60%-80% of the total encoding process [18]. For example, to perform block-based motion estimation for a digitized NTSC video signal with a typical search window size of 15 pixels requires a processing power of about 30 giga operations per second (GOPs). Not to mention other advanced coding features, such as fractional pixel search or rate distortion optimization, these numbers are not feasible for real-time processing even with the dramatic increase of today's CPU speed. However, reducing the motion estimation complexity, for example by limiting the search window, may result in less accurate motion information, which has a high impact on the visual performance. Thus, it is important and challenging to reduce the complexity of video encoder and transcoder while maintaining an acceptable quality.

In addition, the requirement for better media delivery is not only the complexity factor but also the quality factor. Due to the scarce resources and constraints, optimized solutions for video encoder and transcoder sometimes

need to sacrifice the quality. For example, to meet the low channel bandwidth, video bit stream is generally compressed or transcoded using large quantization parameters, which may cause severe loss in visual quality due to the quantization noise. Furthermore, the visual quality of a compressed video is not only influenced by these compression or transcoding techniques, but also by other factors such as data source or channel condition. When the resource is available and constraints are no longer concerned, how to obtain better quality from these reconstructed video becomes important.

To compound this problem, quality enhancement by post-processing has been proposed as one of the potential solutions [19–22]. If the aim of optimized solutions for video encoder and transcoder is mostly to reduce the complexity at the cost of decreasing video quality, post-processing can be considered as a complementary solution to improve the visual quality without the need of changing the encoder or transcoder structure. Together with video coding and transcoding processes, optimized post-processing techniques for quality enhancement have also become an active research area in recent years.

1.2 Thesis Scope and Main Contributions

In this thesis, we are mainly concerned with the optimized techniques for the video compression and transcoding processes as well as the reconstructed video quality of these processes. On the one hand, we explore the solutions to optimize the video encoder and transcoder in terms of complexity reduction. On the other hand, we consider the quality degradation issue, which is affected by these coding and transcoding solutions, in order to enhance the decoded video

by post-processing. Solutions for these related areas will ensure better and faster visual communications services and multimedia applications in terms of efficient storage, transmission, and display. The improvement of video encoder and transcoder allows for efficient storage and transmission among today's heterogeneous networks and devices, and make it suitable for real-time applications. Meanwhile, the optimized post-processing techniques for quality enhancement complement the inadequacy in terms of quality degradation of the decoded or transcoded video for better delivered video quality.

Specifically, we focus on the block-based video coding systems, both existing video coding standards H.263 or MPEG-2 and the state-of-the-art H.264/AVC standard. In this thesis, we analyze the complexity of video encoder and transcoder to provide efficient methods for complexity reduction while maintaining an acceptable visual quality. Due to the advanced technologies employed in the latest standard H.264/AVC, a complete picture of video transcoding between the existing standard such as H.263 and H.264/AVC is presented. Also proposed are efficient post-processing methods to improve the reconstructed video quality from video coding or transcoding process. The works in this thesis focus on the following topics:

1. Fast block-matching algorithm for motion estimation

We first consider the structure of a typical video coding system in order to reduce the complexity while maintaining an acceptable quality. We mainly focus on motion estimation and prediction in our solution as it is one of the most computationally intensive components in a video encoder. We then study many techniques to reduce the computational complexity of the motion estimation and focus on the simplified block-matching measure to

speed up the estimation process. In addition, hierarchical search approach and early termination technique are also employed to use together with the fast block-matching measures.

2. Efficient transcoding method with enhanced rate control

To support universal access over the diversity of networks, devices, and video coding standards in many real-time applications, efficient transcoding methods are required. The latest standard H.264/AVC, which has incorporated many advanced technologies, makes the application of existing transcoding techniques no longer efficient nor suitable. We investigate and focus on format transcoding and downsizing transcoding between existing standard H.263 and H.264/AVC to speed up the process. In addition, we also consider an enhanced rate control when transcoding to H.264/AVC to achieve a better visual quality of the transcoded video.

3. Blind video enhancement from multiple compressed copies

Efficient coding and transcoding techniques for suiting different constraints such as storage, network bandwidth, and available computing resource can achieve notable reduction in the complexity but sacrifice the quality. Here, we mostly focus on post-processing techniques to enhance the quality of the reconstructed video from compressed bit streams. We consider a new research problem by blindly enhancing the quality of the video reconstructed from multiple compressed copies of the same video content. The main objective is to reconstruct a video that achieves better quality than any of the available copies.

The main contributions in this thesis are summarized below:

- New block-matching measures are proposed for a fast evaluation of the match between two blocks in motion estimation. These measures are computed based on the features of the block such as block sum and block variance. Integral frame attributes are proposed to use for fast computation of these block features.
- A fast motion vector re-estimation and an intra-prediction mode selection are proposed. Also proposed is an enhanced rate control based on a model to approximate the relationship between the total number of coding bits and quantization step sizes between the precoded and transcoded videos. This model is used for selecting the quantization parameters at the sequence, group-of-picture, and frame levels.
- The reconstruction of the coefficients of the enhanced video in transform domain is proposed based on the narrow quantization constraint set derived from multiple compressed copies. The Laplacian and Cauchy models are also studied to approximate the distribution for each AC transform coefficient to minimize the distortion of the enhanced reconstructed video.

1.3 Thesis Organization

After the first chapter which provides an introduction of the thesis, in Chapter 2, we give a brief overview of the video coding background and review the related work within the scope of the thesis.

In Chapter 3, we present an efficient block-matching algorithm for motion estimation and compensation. New block-matching measures are described based on the features of the block. The computations of these measures are speeded up

by using the integral frame attribute concept. To further reduce the complexity, hierarchical search approach and early termination techniques are presented.

In Chapter 4, a complete picture of format transcoding and downsizing transcoding between H.263 and H.264/AVC standards is presented. By using the proposed enhanced rate control model for H.264/AVC transcoding, we obtain a better visual quality of the transcoded video. Extensive experiments have been conducted to demonstrate the efficiency and effectiveness of the proposed method.

Chapter 5 presents a framework for blindly enhancing the quality of the video reconstructed from multiple compressed copies of the same video content. A method to reconstruct each coefficient of the video in transform domain by using a narrow quantization constraint set derived from multiple compressed copies is proposed. Analytical and experiment results show that the proposed method can reconstruct a video that achieves better quality than any of the available copies.

Finally, we provide the concluding remarks and future work in Chapter 6.

1.4 Summary

In this chapter, we have provided some background and motivations for the importance of optimized techniques in video compression and enhancement. We have described the thesis scope and summarized the main contributions. An overview of the remainder of the thesis has been given.

Chapter 2

Background and Literature Review

In this chapter, we provide some basic concepts of digital video and a review of the video coding background. The background introduced in this chapter is necessary for understanding the main work in the remainder of the thesis. The literature review of the related work of this thesis is also given.

This chapter is organized as follows. In Section 2.1, we describe some preliminaries of digital video and the principles of video compression. In Section 2.2, we present a brief overview of the video coding standards. After providing the literature review of the related work within the scope of the thesis in Section 2.3, we summarize this chapter.

2.1 Video Compression Fundamentals

2.1.1 Preliminaries

2.1.1.1 Digital Video

Digital video is the representation in the discrete form of a natural (real-world) visual scene. A video sequence can be considered as a sequence of still images (or video frames) representing the scene over a period of time, in which each video image is the 2-D representation of a 3-D scene with varying texture and illumination but no depth information (see Fig. 2.1).

Figure 2.1: Representation of a 3-D scene over a period of time.

Each still image in a video sequence represents the spatial sampling of a real visual scene, which can be obtained by temporally sampling the scene at regular

(a) Progressive sampling

(b) Interlaced sampling

Figure 2.2: Conceptual illustration of the progressive sampling and interlaced sampling of a CIF video frame.

intervals in time. Each spatio-temporal sample is represented in discrete form as a number or set of numbers, which describe the brightness and color information of the sample. A video signal may be sampled either progressively as a series of complete images or interlacedly as a sequence of interlaced fields. In interlaced sampling, half of the information in a complete video frame, which consists of either the odd-numbered (*top field*) or even-numbered (*bottom field*) lines, is sampled at every temporal sampling interval. Fig. 2.2 shows the illustration of progressive and interlaced samplings, in which each band of either top or bottom field in the figure conceptually represents an alternative scanning line.

A color digital video is commonly represented by the tristimulus values of Red, Green, and Blue channels in the RGB primary color space in the systems of video capture and display. However, RGB primary is not necessarily the most efficient representation of color signal for processing and transmission. As the human visual system is less sensitive to color than to luminance (brightness), the popular Y:Cb:Cr color space is more efficient in representing a color video image by separating the luminance from the color information, where Y is the luminance component and Cb and Cr are the two difference chrominance components. The conversion between RGB primary and Y:Cb:Cr color space can be given as follows [23]

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \quad (2.1)$$

and

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.164 & 0.000 & 1.596 \\ 1.164 & -0.392 & -0.813 \\ 1.164 & 2.017 & 0.000 \end{bmatrix} \begin{bmatrix} Y - 16 \\ Cb - 128 \\ Cr - 128 \end{bmatrix} \quad (2.2)$$

Representing a video frame in the Y:Cb:Cr color space can reduce the amount of data by sub-sampling the chrominance components with a lower resolution than the luminance component without having an obvious effect on visual quality. Three popular patterns for sub-sampling Cb and Cr components are shown in Fig. 2.3. In 4:4:4 sampling, the three components (Y:Cb:Cr) have the same resolution, which preserve the full fidelity of the color information. 4:2:2 sampling is commonly used for high color quality representation, where

Figure 2.3: Popular pattern representations of a color video frame in the Y:Cb:Cr color space.

the chrominance components have the same vertical resolution but half the horizontal resolution compared with those of the luminance component. 4:2:0 sampling is popular in many digital video applications such as digital television, DVD storage, and video conference. In 4:2:0 sampling, each color chrominance component contains a quarter of the samples of the luminance component (i.e., for every four samples of Y component there are one Cb and one Cr samples).

2.1.1.2 Video Frame Format

To standardize the digital formats for representing different analog TV video signals, the BT.601 recommendation for TV production, which is developed by the International Telecommunications Union - Radio Sector (ITU-R), is widely used [24]. Two digital video formats for 525-line (NTSC) and 625-line (PAL/SECAM) TV systems are defined with 4:3 and 16:9 aspect ratios, respectively. The parameters of the BT.601 formats are illustrated in Fig. 2.4, in which the active areas have the resolutions of 720×480 pixels (525/60) and

Figure 2.4: BT.601 standard formats for NTSC and PAL/SECAM TV systems.

720×576 pixels (625/50). The 4:3 aspect ratio version of the BT.601 formats is formerly known as International Consultative Committed for Radio, CCIR601 format.

In addition to the BT.601 formats, a number of intermediate digital video formats have been defined for various applications. Table 2.1 summarizes the parameters of these video formats together with their main applications. The CIF (Common Intermediate Format) is specified by International Telecommunications Union - Telecommunications Sector (ITU-T) for the video conference with about quarter the resolution of BT.601, and QCIF with a quarter resolution of CIF is used for video phone type applications [9]. For medium quality video such as CD movies and video games, the SIF (Source Intermediate Format) is widely used, which is about the same as CIF. In addition, to support High-Definition Television (HDTV) video quality, the Society of Motion Picture and Television Engineers (SMPTE) has standardized several HDTV formats for consumer HDTV and studio HDTV (see Table 2.1) [25].

Table 2.1: Parameters of a number of intermediate digital video formats and their typical applications.

Format	Resolution	Applications
Sub-QCIF	128×96	Handheld mobile video
QCIF	176×144	Video telephony over wired/wireless modem
CIF	352×288	Video conferencing over ISDN/Internet
SIF	$352 \times 240/288$	Intermediate quality video distribution
4CIF	704×576	High quality video distribution (DVD, SDTV)
CCIR 601	720×480	TV
HDTV 1440	1440×960	Consumer HDTV
16CIF	1408×1152	
HDTV	1920×1080	Studio HDTV

2.1.1.3 Video Quality Measure

A quality measure is necessary in video processing to gauge the difference between the original and processed videos. For example, to evaluate the performance of a video coding system, it is desirable to measure the distortion caused by compression. One of the most popular objective quality measures in video processing is the mean square error (MSE). The MSE between two video frames at the same time instance t of the video sequences f and g can be defined as

$$\text{MSE} = \sigma_e^2 = \frac{1}{N} \sum_{x,y} (f(x, y, t) - g(x, y, t))^2 \quad (2.3)$$

where N is the total number of pixels in the video frame, $f(x, y, t)$ and $g(x, y, t)$ represent the pixel values at the location (x, y) for the t -th frames of these videos.

Alternatively, the peak-signal-to-noise ratio (PSNR) in decibel (dB), which can be considered as another form of MSE, is widely used as a quality measure

in video coding. This measure is the ratio of peak-to-peak signal to the root mean square error and defined as

$$\text{PSNR} = 10 \log_{10} \frac{\psi_{\max}^2}{\sigma_e^2} \quad (2.4)$$

where $\psi_{\max}$ is the peak (maximum) intensity value of the video signal. For a digital video represented with 8 bits per pixel for each color component, the value of $\psi_{\max}$ is 255. Although MSE or PSNR does not correlate very well with subjective quality measure of the visual signal distortion, these measures are still commonly used due to their mathematical tractability for the development and comparison purposes of the video processing algorithms.

2.1.2 Principles of Video Compression

The basic principle of digital video compression is to reduce the redundant information in a video signal. A typical video encoder generally employs efficient compression techniques to remove the statistical redundancy of the digital video signal in both spatial and temporal domains. Fig. 2.5 shows a typical two-stage process to achieve the compression of a video signal. In the first stage, techniques that exploit the temporal redundancy between adjacent video frames will be employed. The spatial and spectral redundancy in response to the human visual system of the output from the first stage will be removed in the second stage.

Motion-compensated prediction is commonly used in video compression to exploit the temporal redundancy between neighboring video frames. In essence, motion estimation techniques are employed to estimate the spatial-temporal

Figure 2.5: Two basic stages of the video coding process.

Figure 2.6: Motion estimation and compensation block diagram.

variation of intensity to obtain the true motion information between time-varying video frames. The current frame is then predicted based on the motion information and the previous frame using motion compensation and the difference (residue) between the current frame and predicted frame is sent to further remove the spatial redundancy (see Fig. 2.6). A large number of techniques have been proposed to estimate the satisfactory motion information such as block-matching techniques, optical flow techniques, mesh-based techniques, etc. [26–28].

To remove the spatial redundancy between data samples, transform coding is widely adopted. The main goal of transformation is to decorrelate the input data samples and concentrate their energy into a small number of visually significant transform coefficients. There are a number of transformations that exhibit such property such as Karhunen-Loveve transform (KLT), discrete cosine transform (DCT), and discrete wavelet transform (DWT) [2]. However,

Figure 2.7: The block diagram of a generic model of the video encoder and decoder.

the transformation process itself does not achieve any compression as the energy in both spatial and transform domains are equal due to the orthogonality of these transformations. In a lossy compression system, quantization followed by entropy coding is commonly adopted to reduce the data rate by removing the insignificant coefficients to the human visual system. A generic model of the video encoder and decoder is presented in Fig. 2.7. In the following, we shall provide more specific details of these components in a typical video coding system.

2.1.2.1 Block-Based Motion Estimation

A variety of techniques have been proposed to estimation the true motion information between varying temporal samples of a video sequence. However, due to the tractability of computation and ease of hardware implementation, block-matching techniques are commonly adopted in a typical video coding system

Figure 2.8: Block-matching motion search.

for motion estimation. The basic concept of block-based motion estimation is to partition the current frame into a set of non-overlapping blocks. Motion estimation then aims at finding the best-matching block in the reference frame for each block in the current frame, which minimizes the residual energy. As object movement in image sequences is two dimensional, a 2-D displacement (motion vector) is therefore to be found for each block.

Fig. 2.8 depicts the principle of block-matching algorithm. Given a reference frame and an $N_1 \times N_2$ block in a current frame, the objective of motion estimation is to determine the $N_1 \times N_2$ block in the reference frame that better matches (according to a given criterion) the characteristics of the block in the current frame. The location of the block region in the current frame is given by the coordinates (x, y) of their top-left corner. Ideally, we would like to search the whole reference frame for the best match; however, this is usually imprac-

tical due to the large number of comparisons required. Instead, the search is restricted to a $[-W, W]$ search region around the original location of the block in the current frame. Let $(x + u, y + v)$ be the location of the best-matching block in the reference frame. In motion estimation terminology, the vector from (x, y) to $(x + u, y + v)$ is referred to as the motion vector associated with the block under consideration at location (x, y) . Often the motion vector is expressed in relative coordinates; that is, the motion vector is simply expressed as (u, v) .

Matching criteria The accuracy and complexity of motion estimation depend on the matching criterion function applied in motion search. Two conventional criteria are widely adopted as follows:

(1) **The mean square error (MSE):** The motion vector under estimate is to minimize the MSE, defined as

$$\text{MSE}(u, v) = \frac{1}{N_1 N_2} \sum_{(m, n) \in \mathcal{B}} (f_c(m, n) - f_r(m + u, n + v))^2 \quad (2.5)$$

where $\mathcal{B}$ denotes an $N_1 \times N_2$ block for a set of candidate motion vector (u, v) , $f_c(m, n)$ and $f_r(m + u, n + v)$ denote the pixel values in the current frame and reference frame, respectively.

(2) **The mean absolute difference (MAD):** This criterion provides a reasonably good approximation of the residual energy and is easier to calculate than MSE, given by

$$\text{MAD}(u, v) = \frac{1}{N_1 N_2} \sum_{(m, n) \in \mathcal{B}} |f_c(m, n) - f_r(m + u, n + v)| \quad (2.6)$$

In practice, due to the computational simplicity and ease of implementation,

the MAD is often used as the matching criterion and normally referred to as sum of the absolute difference (SAD).

Fractional pixel motion search The aforementioned block-matching procedure can only obtain the motion information with the accuracy in terms of integer pixel displacements. However, a moving object does not necessarily move to the position on the pixel grid, but often between pixels. Thus, by estimating the displacement at a higher resolution (fractional pixel accuracy), we can obtain a better performance for motion estimation.

To perform the fractional pixel motion search, interpolation is required to produce the sample values at finer resolutions. To produce the sample values at half-pixel positions, a simple bilinear interpolation is used in MPEG-1/2/4 or H.263, while one-dimensional 6-tap FIR filter is adopted in H.264/AVC [29]. To generate the half-pixel samples for searching motion vector with half-pixel accuracy, the original video frame needs to be interpolated twice in each vertical and horizontal direction.

Fig. 2.9 illustrates one possible implementation of the fractional pixel motion search. To find the motion vector with half-pixel accuracy, the encoder examines the half-pixel samples around the best match on the integer sample grid and selects the best match, which minimizes the residual energy according to some matching criterion. If the motion vector with quarter-pixel accuracy is required, the encoder continues to search the quarter-pixel samples around the best half-pixel match.

Figure 2.9: Fractional pixel motion search.

2.1.2.2 Transform Coding

A number of transforms have been proposed for video compression, which can be classified broadly into two categories: block-based and image-based [23]. Block-based transform such as DCT and KLT is applied on blocks of $N \times N$ pixels and processes the video frame in units of a block. Image-based transform such as DWT operates on the complete video frame, which has been shown to outperform block-based transform but requires more memory. Due to the good compaction performance over a wide class of visual signals and ease of implementation for fast computation, 2-D DCT is commonly used for video coding, which is defined as [2]

$$\begin{aligned}
 X(k, l) &= \frac{4}{N^2} C(k) C(l) \sum_{m, n=0}^{N-1} x(m, n) \cos\left[\frac{k\pi}{N}(m + 0.5)\right] \cos\left[\frac{l\pi}{N}(n + 0.5)\right] \\
 &\text{for } k, l \in \{0, \dots, N-1\} \\
 &\text{and } C(k), C(l) = \begin{cases} \frac{1}{\sqrt{2}}, & \text{for } k, l = 0 \\ 1, & \text{otherwise} \end{cases}
 \end{aligned} \tag{2.7}$$

Figure 2.10: Basis images of 8×8 DCT transform.

where N is the size of the image block, $x(m, n)$ and $X(k, l)$ denote the values in the spatial and transform domains, respectively. For the reason of practicality, the DCT is usually applied to a small block of the video frame. Fig. 2.10 shows the basis images corresponding to 8×8 DCT, which is adopted in a number of video coding standards such as H.261/2/3 and MPEG-1/2/4.

2.1.2.3 Quantization

To reduce efficiently the data rate of video signals, the quantization process for transform coefficients is employed. The goal of quantization is to remove the transformed data that are less important to the visual appearance of the image and retain the visually important components. It exploits the fact that most of image energy is concentrated at the lower frequencies and the human eyes are less sensitive to the distortion at higher frequencies to achieve better compression through perceptually weighted quantization. In general, scalar

(a) With dead zone, $g = 0$.

(b) Without dead zone, $g = 0.5$.

Figure 2.11: Illustration of a uniform scalar quantization with and without dead zone.

quantization is used in many video coding standards and can be defined as [3]

$$\begin{aligned} Y(k, l) &= \text{sgn}(X(k, l)) \left\lfloor \frac{|X(k, l)|}{\Delta} + g \right\rfloor \\ \hat{X}(k, l) &= Y(k, l) \times \Delta \end{aligned} \quad (2.8)$$

where the function $\text{sgn}(\cdot)$ returns the sign of a signal, Δ is the quantization step size, g is the rounding control parameter, and $\lfloor \cdot \rfloor$ is the floor operator that rounds to the nearest integer towards minus infinity, $Y(k, l)$ and $\hat{X}(k, l)$ are the quantized and dequantized values, respectively. The rounding parameter g controls the position of the decision level that can result in quantizers with or without dead zone (see Fig. 2.11).

2.1.2.4 Entropy Coding

Entropy coding is employed as a statistical compressed technique to efficiently encode the quantized values of transform data, which contain a few significant non-zero coefficients and a large number of zero coefficients. As most of the energy is packed at the low frequencies and high frequency coefficients are generally very small and often quantized to zero, the quantized coefficients are rearranged into 1-D array by an ordered scanning operation for effective entropy coding. Fig. 2.12 shows a typical zigzag scan order in most of the video coding standards for a progressive frame.

The basic principle of entropy coding is to assign the short code words to the highly probable symbols and the long code words to the less probable ones. Thus, the entropy coding process is also referred to as the variable length coding (VLC), in which the lengths of the codes vary inversely with the probability of occurrences of the various symbols. Two types of VLC, which are commonly

Figure 2.12: Zigzag scan order for a progressive video frame.

employed in the video coding standards, are the Huffman coding and arithmetic coding (refer to [4, 5] for more details).

2.1.3 Representations of Compressed Video

2.1.3.1 Layer of Video Data

In general, the highest layer of the coded video is the video sequence, which is a series of one or more pictures. For progressive (non-interlaced) video, a picture corresponds to a single frame of the video sequence, while for interlaced video, a picture can refer to as a frame or the top field or the bottom field of the frame depending on the context.

Pictures in contiguous display order are collected into a group-of-pictures (GOP) for random access purposes. A GOP is a roughly independently decodable sequence of frames. A set of GOPs form a video sequence which is a top coding layer (see Fig. 2.13). Pictures are divided into slices to give some immunity to data corruption. Slices are then partitioned into macroblocks which are

Figure 2.13: The structure of the coding unit.

generally the units for motion estimation and compensation in H.263 or MPEG-2. Every slice shall contain at least one macroblock and slices shall not overlap. In the most general case, it is not necessary for the slice to cover the entire picture, and thus the slice can be as big as the whole picture and as small as a single macroblock. In the case of data corruption, such as transmission errors, a decoder can drop a slice and not the whole picture to allow for a smoother recovery. Barring transmission error, adding slices does not change quality or speed; the only effect is slightly worse compression.

In video compression standards, macroblocks often consist of an array of

16×16 pixels. The choice of macroblock size of 16×16 pixels is a result of the trade-off between the coding gain provided by using motion information and the overhead needed to store it. Blocks are basic coding units and the transformation is applied at this level. There are 8 rows and 8 columns in the block. A block size of 8×8 pixels has been chosen to provide a suitable trade-off in exploiting the correlations and providing adaptations. For 4:2:0 sub-sampling format of color video, each macroblock consists of four luminance blocks followed by one Cb block and one Cr block.

2.1.3.2 Picture Type

There are three basic types of pictures that use different coding methods in most of the video coding standards. An intra-coded picture (I-type picture) is coded using the information only from itself with no reference to any past or future picture. A predictive-coded picture (P-type picture) is a picture which is coded using motion-compensated prediction from a past I-type or P-type picture. A bidirectionally predictive-coded picture (B-type picture) is a picture which is coded using motion-compensated prediction from a past and a future I-type or P-type pictures.

In block-based motion estimation, the best-matching block in the past reference picture is estimated for each block in P-type pictures. The difference (residue) between these two blocks is then transform coded (see Fig. 2.14). For each block in B-type pictures, two best-matching blocks, one from the past reference picture and one from the future reference picture, are obtained (see Fig. 2.15). The encoder can form a block prediction error from either one of these two candidate blocks or their weighted average. Due to the bidirectional

Figure 2.14: Forward motion compensation for P-type pictures.

prediction used in B-type pictures, some reordering of the input pictures is needed at the encoder so that they are coded and delivered in the correct order to the decoder.

One obvious choice of a group of pictures is to have the I-type picture at the beginning of the group followed by P-type or B-type pictures in some order. An example of GOPs for a typical coding scheme is depicted in Fig. 2.16 where each GOP consists of twelve frames allowing for a random access requirement. The GOP length depends on the maximum access time we are willing to allow.

Compared with the coded video sequence of H.261 with only two picture types (I-type and P-type pictures), MPEG-2/4 and H.263/4 include B-picture types to solve the problem that current frame predictions need information in

Figure 2.15: Bidirectional motion compensation for B-type pictures.

the future reference picture. H.263 includes an optional picture type, namely PB-type picture which consists of one P-type picture and one B-type picture. In the latest standard H.264/AVC, two new picture types are introduced, namely SP-type and SI-type. While SP-type picture is coded for efficient switching between different precoded pictures (switching P-type pictures), SI-type picture allows an exact match of a macroblock in an SP-type picture for random access and error recovery purposes (switching I-type pictures). For details on the novel concept of SP and SI-type pictures, the reader is referred to [30].

Figure 2.16: Example of a coded video sequence: (a) MPEG-2 (b) H.261.

2.2 Video Coding Standards

With the importance of video coding and transmission in the 1980s, a number of international video coding standards have been developed and standardized by two standard bodies, the International Standards Organization (ISO) and the International Telecommunications Union (ITU). Each of these standards supports a particular application or a set of applications of video coding. For example, ISO issued standards to support storage and distribution applications by Moving Picture Experts Group (MPEG) such as MPEG-1/2/4 [6–8] while ITU focused on standards to support real-time, two-way video communications such as H.261/3/4 [9–11].

Besides targeting for a particular class of applications, the aim of these stan-

dards is to ensure interoperability between equipments and systems from different manufacturers. Each of these standards only describes a syntax or method of representation for compressed video data and defines the decoding procedure. In what follows, we present the basic architectures of existing standards such as H.263 and MPEG-2 and highlight some key advanced technologies in the latest standard H.264/AVC in order to understand the remainder of this thesis.

2.2.1 H.263 and MPEG-2

Essentially, the generic video coding schemes of most video compression standards such as H.261/2/3 and MPEG-1/2/4 are somewhat similar. Fig. 2.17 shows the typical high-level structures of the encoder and decoder for these standards. The model consists of four main stages: a motion estimation and compensation stage, a transformation stage, a lossy quantization stage, and a lossless entropy coding stage.

Both standards generally share common coding features of a typical block-based hybrid video coding system. For example, a block-matching algorithm is employed to obtain satisfactory motion vectors with the half-pixel accuracy for motion-compensated prediction. Meanwhile, the 8×8 DCT is applied in the transformation stage followed by a simple scalar quantization and a variable length coding. However, there are several negotiable advanced coding features in H.263 which make it different from MPEG-2.

One of the main differences is the advanced prediction mode supported by the H.263 standard. This mode allows for the use of four motion vectors in a macroblock, one for each of the four 8×8 blocks. It provides a greater flexibility in obtaining a best match for the macroblock. Furthermore, overlapped

(a) Encoder

(b) Decoder.

Figure 2.17: Block diagrams of encoder and decoder for MPEG-2 and H.263.

block motion compensation can be used and the motion vectors are allowed to point outside the picture for better prediction of the movement around the picture boundaries, which is referred to as the unrestricted motion vector option. Besides the conventional I, P, and B picture types, H.263 incorporates an additional PB-type picture to improve temporal resolution, which consists of a P-type picture and a B-type picture. In addition, it supports a syntax-based arithmetic coding mode, which is also lossless coding but can further reduce the bit rate.

2.2.2 H.264/AVC

The H.264/AVC video standard incorporates a set of new coding features to achieve higher coding efficiency. Fig. 2.18 shows the typical structure of the H.264/AVC encoder.

Figure 2.18: Block diagram of a H.264/AVC encoder.

Figure 2.19: 4×4 luminance prediction modes. Figure 2.20: Intra 16×16 prediction modes.

One of the new advanced techniques employed in H.264/AVC is the intra prediction, in which a block is predicted from the previously decoded information of the neighboring areas within the current frame. For the luminance samples, the prediction block is formed for either a 4×4 block or a 16×16 macroblock, which is interpolated from neighboring reconstructed samples in a number of directions. There are a total of nine optional prediction modes for each 4×4 luminance block, four modes for a 16×16 luminance block, and four modes for the chroma components. Figs. 2.19 and 2.20 indicate the directions

Figure 2.21: Variable block sizes for H.264/AVC motion estimation.

of prediction in each mode for 4×4 block and 16×16 macroblock, respectively.

In addition to the new intra macroblock coding type, variable block-size motion compensation with small block sizes is also supported in H.264/AVC. The luminance component of each 16×16 macroblock can be partitioned into 16×16 , 16×8 , 8×16 , and 8×8 samples, and when necessary, each 8×8 block of samples can be further partitioned into 8×4 , 4×8 , and 4×4 samples, resulting in a combination of seven motion-compensated prediction modes. Fig. 2.21 illustrates these different block modes for motion prediction in H.264/AVC.

To attain more precise motion compensation in areas of fine or complex motion, the motion vectors are specified in quarter-pixel accuracy. In addition, the new design of H.264/AVC extends the enhanced reference picture selection techniques in H.263++ for higher coding efficiency by allowing a large number of reference frames for motion compensation purpose.

Other important and new features of H.264/AVC include in-loop de-blocking

filter, 4×4 integer transform, context-adaptive variable length coding (CAVLC) and context-adaptive binary arithmetic coding (CABAC), etc. Interested readers can refer to [11] for more details.

2.3 Related Work

In this section, we provide the literature review of the related work within the scope of the thesis. Here, we mainly focus on the optimized techniques for video encoder and transcoder for better performance in terms of complexity and quality. Our major concern for video compression and transcoding processes is complexity reduction, while quality enhancement by post-processing is the other main interest to improve the quality of the decoded and transcoded videos.

2.3.1 Block-Matching Motion Estimation

Motion-compensated prediction is one of the key components for efficient inter-frame coding. In the literature of video compression, a large number of methods have been proposed to estimate satisfactory motion vectors for motion-compensated prediction. Although these methods can effectively achieve the optimal motion displacements, the estimation process is rather computationally intensive that makes it challenging for real-time encoder. This issue has attracted much research interest for optimizing the video encoder in terms of complexity reduction. Most of the works focus on block-based motion estimation, which is widely used in video coding standards.

Full Search Given the matching criterion such as the SAD, the simplest method to find the motion vector for each block in the current frame is to match with all the candidate blocks within a search window of a reference frame. This is referred to as the full search (FS) algorithm. Being exhaustive, the FS algorithm achieves optimal performance in terms of matching cost, but it requires a huge amount of computations, which makes it unsuitable for many real-time applications. Thus, many fast techniques have been proposed to obtain sub-optimal matches with lower complexity. Research in this area can be categorized as follows:

Fast Search by Limiting Search Point These methods obtain sub-optimal matches by limiting the number of search locations in the reference frame. The choice of the limited search locations is based on some heuristic criterion. Koga *et al.* [31] proposed the three-step search (TSS) in 1981, which has become the most popular fast search scheme for low bit rate video applications and is recommended by several standards [9, 10].

The principle of TSS is illustrated in Fig. 2.22. In the first step, TSS examines 9 search locations that are equally distributed on the search window and computes the SAD at each search location. The initial distance between the search locations for a search window of $\pm W$ is given by $d_1 = 2^{k-1}$, where $k = \lceil \log_2 W \rceil$. This initial distance d_1 is a measure of how sparse the motion search is. With a fixed number of search points, the more they are apart from each other, the larger area, which translates to larger motion vectors, they will cover on the search window. In the TSS algorithm, the second step is performed around the winning point of the first step. The search scale is halved due to the reduction of the search point distance by a factor of 2. The same operation is

Figure 2.22: Three-step search algorithm.

repeated in next steps except that the search point distance is further reduced by half for each step. The search process is terminated when the search point distance is equal to 1.

In addition to the TSS algorithm, many proposed various fast search algorithms using the similar approach, such as the new three-step search (NTSS) [32], four-step search (4SS) [33], and block-based gradient descent search (BBGDS) [34] algorithms. These algorithms often employ the squared-shape search pattern of different sizes with a limited number of search steps. More recently, many fast algorithms have been proposed based on the unprecedented suggestion of unrestricted search steps and non-rectangular search patterns, including diamond search (DS) [35–37] and hexagon-based search (HEXBS) [38].

By examining fewer search points, these algorithms can yield speed improvement over the squared-shape search patterns while obtaining similar distortion.

One of the main drawbacks of these algorithms is to assume that the match of two blocks, gauged by some block-matching measure, decreases monotonically as the search location moves away from the best match (i.e., the one with the optimal block-matching measure). Obviously, this assumption essentially requires that the error surface be unimodal over the search window. Unfortunately, this is usually not true due to many reasons such as the aperture problem, the textured (periodical) local image content, the inconsistent block segmentation of moving object and background, the luminance change between frames, and thus the error surface is often non-unimodal. As a consequence, the search for a global minimum would easily be trapped into a local minimum.

Fast Search by Spatio-Temporal Correlation Due to the continuity of the moving object surface and the fact that each object consists of a number of blocks, often motion vectors are highly spatially and temporally correlated. By taking advantage of this property, a motion vector can be estimated among the motion vectors from the adjacent blocks or neighboring frames. Liu and Zaccarin [39] presented a fast search algorithm based on subsampled motion field estimation. The proposed algorithm finds the motion vectors of only some of the blocks in the video frame by using a highly accurate block-matching technique. The motion vectors of the other blocks are estimated from those of the neighboring blocks.

In addition, many have exploited this correlation property to obtain the initial motion vector predictor and propose to search around this predicted motion vector [40–42]. For example, the motion vector field adaptive search (MV-

FAST) in [41] considers a set of predictors by taking the zero motion vector and the motion vectors of the three spatially adjacent blocks (left, top, top-right). Meanwhile, the predictive motion vector field adaptive search (PMVFAST) [42] improves upon MVFAST by proposing generalized predictor selection. The experimental results show that by using the predicted motion vector, these methods can achieve notable complexity reduction without degrading the quality.

Fast Search by Fast Block Comparison These methods reduce the computational cost of a block-matching algorithm by simplifying the block-matching measure evaluated at each search location. One of the key techniques is to reduce the number of pixels used in the computation of the block-matching measure. Specifically, Liu and Zaccarin [39] proposed an alternating pixel decimation scheme for computing the block-matching measure based on a set of pixel patterns such as the check-board pattern. With the decimation ratio of 4 or 2, the block comparison can be 4 or 2 times faster than the existing SAD. The decimation ratio, however, could not be too high. In fact, even with the ratio of 4 there is an obvious degradation in visual quality.

Gharavi and Mills [43] proposed the pixel difference classification (PDC) method, in which each pixel in the block is classified as either the matching pixel or the mismatching pixel according to a predetermined threshold. The more number of matching points, the better the block matching is. The problem with this scheme is the lack of criterion to set up the threshold.

Similarly, Yong *et al.* [44] represented the current and reference frames in binary format using some threshold and applied the conventional search algorithms. Although the computational complexity is reduced significantly, the method degrades the video quality by almost 1-dB in PSNR compared with

the FS algorithm. Alternatively, Song *et al.* [45] proposed to reduce memory bandwidth for efficient hardware implementation by reducing the number of bits per pixel used for SAD calculation, which is referred to as bit truncation. Experimental results show that with 4 bits truncation, the block matching still gives reasonable performance.

Fast Search by Hierarchical Techniques The basic principle behind these techniques is to separate the estimation process into several levels, in which the number of levels is fixed. The motion vector is first obtained at the coarse level using some fast search and then refined gradually in subsequent levels to obtain more accurate motion displacement. One of the particular examples for this category is the multi-resolution motion estimation methods [46–48]. In these methods, the motion estimation process is performed at different resolution representations of the video frame using different search strategies. One way is to use a variable search area at each level. For example, the motion vector is obtained from a large search area at the coarse resolution and the candidate becomes the search center at the higher resolution, which has a smaller search area. By employing the spatio-temporal correlation techniques to estimate the initial motion vector predictor, these methods can further speed up the estimation process.

Fast Search By Fast Computation Techniques The goal of these techniques is to reduce the computational complexity by removing many of the non-optimal search positions by using some criterion without losing the optimality of the block matching in either the conventional FS or fast search algorithms. One of the proposed techniques is to compute the lower bound for each candidate

and compare to the value of the best-matching error so far. If the lower bound is greater than the current best-matching error, the candidate will be removed. As the computation of the lower bound generally requires low complexity, useless computation from impossible candidates can be avoided efficiently without losing the optimality of the best-matching motion vector. One of the pioneer works in this category is the successive elimination algorithm (SEA) [49]. In addition, a number of algorithms modified upon SEA are also reported [50, 51]. These algorithms make use of the sum norm of the macroblock or sub-block as the lower bound, which can be computed efficiently, to compare with the current best-matching error.

In addition, the partial distortion elimination (PDE) technique is also proposed to compute the block-matching measure more efficiently [52]. As the block-matching measure is generally computed as the SAD of a set of representative pixels, PDE technique is proposed to stop the computation halfway if the accumulated partial distortion is greater than the current minimum distortion. The speed-up performance in the PDE algorithm depends on how fast computation of the block-matching measure is stopped according to the partial sum of the matching measure. In [53, 54], the modified algorithms based on PDE are proposed by using different matching scans of the representative pixels or hierarchical PDE in computation of the block-matching measure. Like PDE, the performances of these modified algorithms cannot be analyzed quantitatively and have shown different improvements in terms of the number of computation savings compared with the original PDE for different video contents.

2.3.2 Video Transcoding

Transcoding has played an important role in multimedia communications and services in today's heterogeneous networks and devices. Many have proposed efficient methods to convert one compressed video to another for better suit new application constraints. Bit rate reduction [16, 55], spatial/temporal resolution adjustment [17, 56], and syntax conversion [15, 57] are some examples of common transcoding operations. In addition, video transcoding can perform additional functions like logo insertion [58] or error resilience transcoding [59]. Transcoding between compressed videos of the same standard is generally referred to as homogeneous transcoding, while heterogeneous transcoding provides the conversion between different standards.

Figure 2.23: Architecture of the fully cascaded decoder and encoder transcoder.

The most straightforward solution for video transcoding is to decode the pre-coded video, perform any necessary intermediate processing, and fully re-encode the processed video to meet any new constraints (see Fig. 2.23). Although effective, this fully cascaded decoder and encoder approach is rather computationally intensive. Thus, much research has focused on optimizing the transcoder in order to achieve considerable complexity reduction while maintaining acceptable quality. In the following, we review some of the key techniques to reduce the complexity of the video transcoder in the literature.

Figure 2.24: Architecture of an open-loop transcoder.

Video Transcoding Architecture The simplest transcoding architecture is an open-loop system, whose the complexity is arguably the least intensive. Fig. 2.24 shows the architecture of a typical open-loop transcoder. The transcoder operates based on the principle of either selective transmission such as discarding the high frequency data [60] or requantizing the motion-compensated residue to reduce the bit rate [55]. Without feedback loop, the open-loop system suffers from a serious drift problem due to the mismatch between the reference frames in the decoder and encoder. A number of studies have been conducted to analyze the drift problem for requantization transcoding [61, 62].

To alleviate this problem, the closed-loop transcoder containing a feedback loop to compensate the drift has been proposed [62–64]. Fig. 2.25 shows the architecture of a closed-loop cascaded pixel-domain transcoder. In cases where intermediate processing of the decoded video is not required (e.g., no spatial or temporal resolution reduction), this cascaded pixel-domain architecture can be further simplified by removing one reconstruction loop at the decoder end.

Furthermore, exploiting the linearity of the DCT/IDCT the cascaded pixel-domain transcoder can be simplified to be functionally equivalent to the frequency-domain transcoding architecture [65–68], as shown in Fig. 2.26. In this architecture, motion compensation needs to be performed in frequency do-

Figure 2.25: Architecture of the cascaded pixel-domain transcoder.

main, which can be done by using the methods proposed by Chang *et al.* [65]. In addition, the frequency-domain downscaling techniques [69] may also be employed for downsizing transcoding. Although being able to reduce the computational complexity, this architecture may suffer from the drift problem due to the nonlinear operations including subpixel motion compensation and DCT coefficient clipping. Moreover, motion compensation in frequency domain generally involves floating-point matrix multiplication, which is rather costly. To eliminate some of these issues, Assuncao *et al.* [62] proposed to approximate the floating-point elements by power-of-two fractions to make use of shift op-

Figure 2.26: Cascaded frequency-domain transcoding architecture.

erations, while matrix decomposition techniques [70] have been examined to achieve simplifications. In addition, motion estimation in frequency domain has also been studied [71].

Motion Vector and Mode Mapping Besides employing a variety of transcoding architectures, many have focused on using the side information from the precoded video (e.g., motion vectors, quantization parameters, picture types, or bit allocation statistics) to speed up the transcoding process. One of the key techniques is to reuse the motion vector information in the precoded video for reducing the computational complexity of the motion estimation process [72–76]. The motion vectors in the precoded video are used to map or re-estimate the required motion vectors in the transcoded video. In the case of one-to-one mapping (i.e., the motion vector of a block in the precoded video is corresponding to one motion vector of another block in the transcoded video), motion vector refinement is proposed to refine the motion vectors in the precoded video within a small search window to obtain more accurate motion

information [76,77]. In the case of many-to-one mapping, for example, in downsizing transcoding or syntax transcoding between H.264/AVC with the variable block size motion-compensated prediction feature and H.263, the motion vectors cannot be reused directly and motion vector composition is required.

A number of methods have been proposed to compose a single motion vector from multiple input motion vectors. These include random [78], mean [56], weighted average [72, 79], weighted median [57, 80], and DCMax [81] methods. In the weighted average method, the motion vector in the transcoded video is re-estimated by taking the weighted average of the incoming motion vectors, where the weighting factors can be obtained based on the measure of the corresponding macroblock complexity. For example, Shen *et al.* [79] used the spatial activity measured by the product of the quantization factors and the number of bits used to code. Alternatively, the overlapping area between the macroblock in the precoded video and the final downsizing macroblock is employed as the weighting factor in arbitrary downsizing transcoding [75]. Meanwhile, the median method estimates the motion vector by computing the Euclidean distances between each incoming motion vector, which can yield a good performance. However, directly using the new motion vectors re-estimated by these methods may still introduce considerable quality degradation. To achieve better accuracy, motion vector refinement techniques are proposed at the encoder end [74, 82, 83].

In temporal resolution reduction transcoding, the target frame rate of the transcoded video is generally obtained by skipping frames in the precoded video. As a consequence, the motion vectors that point to the skipped frames in the precoded video are no longer valid. Thus, re-estimation for the motion vectors

of the transcoded video needs to take into account these invalid motion vectors. Hwang *et al.* [73] proposed a bilinear interpolation method to estimate the motion vector from the current frame to the previous nonskipped frame using the motion vectors between neighboring frames. In addition, dominant vector selection techniques are also proposed in [57, 76, 81]. Youn *et al.* [76] re-estimated the invalid motion vector by adding the motion vector of the block in the skipped frame, which has the largest overlapping region with the block pointed by the motion vector of the current block. Alternatively, Chen *et al.* [81] selected the dominant motion vector based on the measures of block activity such as the number of non-zero DCT coefficients or sum of the absolute values of the DCT coefficients. Together using motion vector refinement, the required motion vectors for the transcoded video can be obtained with much lower computational complexity. In addition, to maintain the smooth playback, the frame rate control methods based on the characteristics of the video content have been proposed to adaptively select the number of skipped frames [84, 85].

In addition to motion vector mapping, the conversion of macroblock modes is also considered in video transcoding to obtain higher performance efficiency. The macroblocks in the precoded video, which are coded in different modes, may be no longer suitable nor valid for the transcoded video. For example, in downsizing transcoding, one macroblock in the transcoded video is often corresponding to several macroblocks in the precoded video. The issue also exists in the case of syntax conversion between the existing standards such as H.263 or MPEG-2 and the latest H.264/AVC standard, which supports a number of advanced coding modes. Research that addresses this problem for a variety of transcoding operations can be found in [86, 87]. The basic concept behind these methods is to exploit the useful information of macroblock modes

in the precoded video to select the appropriate coding modes for the macroblocks in the transcoded video.

2.3.3 Quality Enhancement by Post-processing

Optimizing both the complexity and perceptual quality of the video encoder or transcoder is challenging if not impossible as these measures are often traded off against each other. In addition, the perceptual quality is influenced not only by the compression or transcoding method, but also by other factors such as data source, channel condition, or application constraints. Thus, post-processing is considered as one of the promising and complementary solutions to enhance the reconstructed video without the need of changing the encoder or transcoder structure.

There are a large number of post-processing techniques for enhancing the perceptual quality that is degraded by different reasons. Reviewing all of these techniques is impossible and beyond the scope of this thesis. Here, we provide a brief review of the post-processing techniques for quality enhancement, which mainly aim at removing the artifacts caused by compression methods.

In a lossy video compression system, quantization is the major source that introduces the visual quality loss to the original video signal. Together with the block-based coding approach, the error introduced by the quantization/dequantization process may cause undesirable coding artifacts in the reconstructed video at moderate to low bit rates such as blocking artifact, ringing noise, or corner outliers. Many post-processing techniques have been proposed to reduce these coding artifacts of block-based coding. In essence, these techniques can be categorized based on two different points of view as: 1) im-

age/video enhancement based techniques, and 2) image/video restoration based techniques.

Image/Video Enhancement Based Techniques For the post-processing techniques based on image/video enhancement, the goal is to improve the perceived quality subjectively based on the special structure of artifacts and the characteristics of human visual sensitivities. Instead of restoring the pixel back or close to its original value, this approach aims at smoothing visible artifacts to match the perception of the human visual system.

One of the techniques is post-filtering, which applies low-pass filtering to the artifact region (e.g., at high frequency transform coefficients). A linear space-invariant low-pass filtering is first proposed to reduce blocking artifacts in image coding [88–90]. However, the low-pass filter generally blurs the edge details and degrades the visual quality of the image. To solve the problem, a number of sophisticated nonlinear space-variant filtering techniques have been proposed by considering the local image statistics [91–96]. These methods classify image blocks into different categories based on the edge information or local variances and apply various spatial filters for different categories to remove coding artifacts. Together with adaptive spatial filtering, also proposed is temporal filtering to reduce the temporal artifacts such as flickering and motion jerkiness by smoothing the noisy image sequences along motion trajectories [97–99].

Besides the post-filtering techniques, Ismaeil *et al.* [100] proposed to predict the low frequency AC coefficients from the DC coefficients in the current and neighboring blocks to reduce the blocking artifacts in compressed image. The method can provide a good performance in the smooth area, but is unreliable in the neighborhood of edges. Furthermore, Macq *et al.* [101] proposed a

post-processing scheme based on the characteristics of the human visual system, in which a stimulus is not perceived if the contrast value is below the visibility threshold. The proposed method decomposes the compressed image into a number of perceptual channels in order to remove the unmasked artifacts by some visibility threshold corresponding to each channel.

Image/Video Restoration Based Techniques For the post-processing techniques based on image/video restoration, post-processing is formulated as an ill-posed image/video recovery problem. The reconstruction of visual signal is obtained based on the prior knowledge of the original signal as well as the observation of the available data at the decoder side. Basically, these techniques can be classified into the following categories: 1) criterion-based technique, 2) constrained optimization technique, 3) constraint-based technique.

The basic principle of the criterion-based techniques is to recover the artifact image or video in order to meet some predefined optimality criterion. In [102, 103], the linear minimum mean square error (LMMSE) criterion is used to reduce coding artifacts in compressed video. The experimental results show that the LMMSE estimator can provide a reasonable trade-off between artifact reduction and detail preservation. To avoid blurring, the proposed deblocking method in [103] is only applied to the pixels at block boundaries. A linear estimator is also proposed by Guleryuz [104] to improve the reconstructed image by identifying and removing the quantization artifact. The proposed method is based on the models for image and for quantization error to derive the linear estimators of the original image that are optimal in the mean-square-error sense for the worst-case cross correlation between the original and quantization error. Alternatively, Yang *et al.* [105] proposed another nonlinear estimator based on

soft thresholding in lapped transform domain.

In addition to linear and nonlinear estimators, a number of research works have used the maximum a posterior (MAP) criterion to maximize the probability density of the actual visual signal based on the observation of the available data. To reduce the coding artifact, the Huber minimax function is used [106, 107], which is able to smooth artifacts while preserving the edge details. Meanwhile, Li *et al.* [108] proposed a MAP-based method that adopts a nonstationary Huber-Markov model, in which the maximization of the posterior function is obtained by using the random field model. A multiscale MAP method is then proposed in [109] to enhance the reconstructed image from coarse to fine scales. The experimental results have shown that the method can achieve a better performance than the single scale MAP method in terms of suppressing ringing artifacts and reducing complexity.

Unlike the criterion-based techniques, the class of constrained optimization techniques aims at optimizing an optimality criterion subject to some constraints, which are obtained from a prior knowledge of the original visual signal. Yehong *et al.* [110] proposed to use the constrained least square (CLS) method to enhance the blurred image by decomposing into the eigen-face subspaces. Similarly, Hong *et al.* [111] proposed to incorporate CLS regularization in sub-band decomposition and reconstruction to remove the blocking artifact.

In addition, a number of research works have been conducted based on the concept of the constraint-based techniques, which set a number of constraints on the compressed image or video and restore them accordingly. In general, the constraint sets are formed from not only the transmitted data, but also the prior knowledge about the original signal. Many have proposed to reconstruct the

compressed image for the reduction of blocking artifacts based on the popular theory of projection into convex set (POCS) [112–116]. In this approach, a decoded image is reconstructed by the iterative projections onto several constraint sets. These methods generally impose two types of constraint sets: smoothness constraint set (SCS) and quantization constraint set (QCS). The SCS takes the advantage of the strong correlation among the neighboring image pixels to capture the smoothness property of the desired image. On the other hand, QCS is to ensure the fidelity of the processed image not to diverge from an original image due to the projection onto other constraint sets.

Much research has been conducted to design projectors onto the SCS for more efficient reduction of artifacts [117–119]. However, one of the main drawbacks of these methods is their high computational complexity. To address such drawback, a number of methods have been proposed to reduce the complexity per iteration or/and to speed up the convergence process. Kawak and Haddad [120] proposed to reduce the complexity of the method in [110] by cancelling out the DCT and IDCT pairs in each iteration. Alternatively, Lai *et al.* [121] speeded up the iteration process by improving the convergence rate with the frequency-domain filtering. However, compared with other techniques, the complexity of the constraint-based techniques is still costly. Furthermore, the selection of the proper constraint sets for a variety of artifacts is also a challenging task.

2.4 Summary

In this chapter, we have provided some basic concepts of digital video and the video coding background. We have also presented a brief overview of the video coding standards and a review of related work within the scope of the thesis.

Chapter 3

Motion Estimation Based on Integral Frame Attributes

In this chapter, we focus on the improvement for video encoder to reduce the complexity of the encoding process while maintaining acceptable video quality. A new block-matching algorithm is proposed to speed up the motion estimation process based on the concept of integral frame attributes. We also consider the hierarchical approach and early termination technique to attain further computation savings.

This chapter is organized as follows. After presenting an introduction in Section 3.1, Section 3.2 defines the integral frame attributes. In Section 3.3, we describe the proposed block-matching algorithm and how integral frame attributes can be used to reduce the computational complexity of different block-matching measures. Section 3.4 extends the use of the proposed block-matching measures in a two-step approach or a partial summation elimination scheme to further speed up the motion estimation process. The experimental results of

the proposed algorithm are presented in Section 3.5. In Section 3.6, we provide the summary of the main contributions in this chapter.

3.1 Introduction

Due to the tractability of computation and ease of hardware implementation, block-based motion estimation [122] has attracted the most attention in the literature of video compression to estimate satisfactory motion vectors for motion-compensated prediction. In many popular video compression standards, such as H.261/3/4 and MPEG-1/2/4, a block-matching algorithm (BMA) is used to obtain the motion vector for each block. The most widely used BMA is probably the full search (FS) algorithm using sum-of-absolute differences (SAD) block-matching (BM) measure, an algorithm that estimates the motion vector for each block by measuring the SADs with all the candidate blocks within a search window of a reference frame. Being exhaustive, the FS algorithm achieves optimal performance in terms of matching cost, but its high computational requirement makes it unsuitable for many real-time applications.

To reduce the computational complexity, many existing fast BMAs obtain sub-optimal matches by limiting the number of search locations in the reference frame. These include the three-step search (TSS), new three-step search (NTSS), four-step search (FSS), and diamond search (DS), and cross-diamond search (CDS) [31–33, 37]. These BMAs assume that the match between two blocks, gauged by some BM measure, degrades monotonically as the search location moves away from the best match (i.e., the one with the optimal BM measure). As a consequence, these BMAs share a common problem—being

trapped in a locally optimal match and obtaining sub-optimal compression performance.

Instead of limiting the number of search locations, another means to reduce the complexity of a BMA is by simplifying the BM measure evaluated at each search location. One of possible ways is to evaluate the BM measure based on a set of representative pixel patterns for each block. Wang *et al.* [123] defined an adaptive pixel decimation by selecting pixels that are found to have important features in determining a match. Meanwhile, Kim *et al.* [124] proposed an adaptive matching scan and representative pixels obtained by Taylor series expansion. In addition, a binary matching measure based on two-bit transform for motion estimation is proposed in [125, 126]. The proposed method converts image frames into two-bit representations by a simple block-by-block two-bit transform based on multithresholding with mean and linearly approximated standard deviation values. Similarly, Wu *et al.* [127] proposed a DCT-based adaptive thresholding algorithm to achieve an effective binarization of video image for binary motion estimation. The experimental results show that the proposed method can achieve good performance and has lower complexity that is suitable for VLSI implementation with a smaller hardware overhead.

Alternatively, the simplified BM measure can be obtained by representing the current block and each candidate block by a set of block features. These features are used as the matching parameter in the BM measure to reduce the number of computations. In [128–131], horizontal or vertical projections of pixel values in blocks are proposed as the representative block features. These projections are defined as sums of the graylevels along a certain direction of the image and used to compute the BM measure. By exploiting the adjacent

property of these projections, a fast method to compute all the projection values in advance is proposed to reduce the computational complexity. The work shares some similarity with our proposed scheme. However, as their projection values and the derived BM measures are still rather computationally intensive, the existing projection methods can only reduce the complexity by a factor equal to half of the block size (i.e., a factor of 8 for the typical 16×16 blocks). In comparison, the method proposed in this chapter can reduce the complexity by factors of 21-96 for a typical search window size, as we shall show later.

In this chapter, we propose a fast BMA that makes use of features such as block sum and block variance to compute the BM measure. Specifically, we partition each block under consideration into a number of sub-blocks arranged in a fixed pattern, and match the sum or variance of intensity in each sub-block with that of each candidate block. Inspired by the work of Viola *et al.* [132], we first compute the integral attributes of each frame to allow for very fast computation of the proposed BM measures. Although fast computation for block sums had been proposed in [49, 50], our proposed scheme for computing the BM measures can achieve a lower complexity and can be easily extended to other block features as well as fast search algorithms. When used in conjunction with the FS or a fast search algorithm, our proposed BM measures can achieve compression performance very close to that using the existing SAD BM measure, but incurring a much lower computational cost. In addition, the proposed BM measures can be easily integrated into a two-step approach using different block patterns in each step, or used in a partial summation elimination scheme to reject impossible candidates as soon as possible and further reduce the computational complexity.

3.2 Integral Frame Attributes

Given a video frame, let $g(m, n)$ be a frame attribute characterizing some measure of frame features about pixel (m, n) . The measure could be made at a specific pixel (e.g., the pixel's graylevel), or over a neighborhood region of pixels. The integral frame attribute at pixel (m, n) , denoted as $\mathcal{I}_g(m, n)$, is defined as the sum of the frame attributes $g(m, n)$'s over the region that is above and to the left of pixel (m, n) , inclusive [132]; that is (see Fig. 3.1 for illustration)

$$\mathcal{I}_g(m, n) = \sum_{x=0}^m \sum_{y=0}^n g(x, y) \quad (3.1)$$

Let $\mathcal{R}_g(m, n)$ denote the cumulative row sum of frame attributes $g(m, n)$'s, defined as

$$\mathcal{R}_g(m, n) = \sum_{x=0}^m g(x, n) \quad (3.2)$$

Assuming $\mathcal{R}_g(-1, n) = 0$ and $\mathcal{I}_g(m, -1) = 0$, one can compute the integral frame attribute $\mathcal{I}_g$ in one pass by using two recursive formulas:

$$\begin{aligned} \mathcal{R}_g(m, n) &= \mathcal{R}_g(m-1, n) + g(m, n) \\ \mathcal{I}_g(m, n) &= \mathcal{I}_g(m, n-1) + \mathcal{R}_g(m, n) \end{aligned} \quad (3.3)$$

Hence, for a frame with $M \times N$ pixels, only $2MN$ additions are required to compute the complete integral frame attributes in addition to the computational cost required for each frame attribute $g(m, n)$.

Using this integral frame attribute, the sum of the frame attributes in any

Figure 3.1: The value of integral frame attribute at pixel (m, n) is equal to the sum of some frame features over the region that is above and to the left of pixel (m, n) , inclusive, in the original frame.

rectangular block (hereafter referred to as block sum $(\mathcal{BS}_g)$ for simplicity) can be computed with 3 arithmetic operations (1 addition and 2 subtractions). This can be seen from Fig. 3.2, where the $\mathcal{BS}_g$ of block D with support $\Omega_D = \{(x, y) : r < x \leq m, s < y \leq n\}$ can be computed by

$$\begin{aligned} \mathcal{BS}_g(D) &= \sum_{x=r+1}^m \sum_{y=s+1}^n g(x, y) \\ &= \mathcal{I}_g(m, n) - \mathcal{I}_g(r, n) - \mathcal{I}_g(m, s) + \mathcal{I}_g(r, s) \end{aligned} \quad (3.4)$$

using the four corresponding integral frame attributes at the corners of the block. The technique for fast block sum computation in (3.4) is also known as the inclusion-and-exclusion principle in combinatorics and has been widely applied to image coding, segmentation, and color quantization [133–135].

Consider the case where $g(m, n)$ is the graylevel of pixel (m, n) , denoted as

Figure 3.2: The sum of all frame features in block D can be computed by using the four corresponding integral frame attributes at the block boundaries.

$f(m, n)$, then the $\mathcal{BS}_g$ of a rectangular block D becomes the sum of graylevel values in the block (hereafter referred to as $\mathcal{BS}_f(D)$), and it can be computed by

$$\begin{aligned}
 \mathcal{I}_f(m, n) &= \sum_{x=0}^m \sum_{y=0}^n f(x, y) \\
 \mathcal{BS}_f(D) &= \sum_{x=r+1}^m \sum_{y=s+1}^n f(x, y) \\
 &= \mathcal{I}_f(m, n) - \mathcal{I}_f(r, n) - \mathcal{I}_f(m, s) + \mathcal{I}_f(r, s)
 \end{aligned} \tag{3.5}$$

Another block feature that can be easily computed by using integral frame attributes is the sum of the squares of all pixel values in a block, referred to as squared block sum. In this case, $g(m, n)$ is the square function of the graylevel at pixel (m, n) and the squared block sum, denoted as $\mathcal{BS}_{f^2}$, can be obtained

by

$$\begin{aligned}
\mathcal{I}_{f^2}(m, n) &= \sum_{x=0}^m \sum_{y=0}^n f^2(x, y) \\
\mathcal{BS}_{f^2}(D) &= \sum_{x=r+1}^m \sum_{y=s+1}^n f^2(x, y) \\
&= \mathcal{I}_{f^2}(m, n) - \mathcal{I}_{f^2}(r, n) - \mathcal{I}_{f^2}(m, s) + \mathcal{I}_{f^2}(r, s) \quad (3.6)
\end{aligned}$$

Note that one multiplication is required to compute the square of a graylevel value. Assuming one multiplication requires three arithmetic operations [72], a total of $5MN$ arithmetic operations are required to compute the integral frame attribute $\mathcal{I}_{f^2}$. Although the assumption is typically made for some specific processors, less computation resources and cost may be needed with the advanced CPU technology and/or specific hardware designs.

Using the block sum and squared block sum features, we can obtain the variance of all pixel values in a rectangular block D by

$$\begin{aligned}
\delta^2(D) &= \frac{1}{N_D} \sum_{x=r+1}^m \sum_{y=s+1}^n \left[f(x, y) - \frac{1}{N_D} \sum_{x=r+1}^m \sum_{y=s+1}^n f(x, y) \right]^2 \\
&= \frac{1}{N_D} \sum_{x=r+1}^m \sum_{y=s+1}^n f^2(x, y) - \left[\frac{1}{N_D} \sum_{x=r+1}^m \sum_{y=s+1}^n f(x, y) \right]^2 \\
&= \frac{1}{N_D} \mathcal{BS}_{f^2}(D) - \left[\frac{1}{N_D} \mathcal{BS}_f(D) \right]^2 \quad (3.7)
\end{aligned}$$

where N_D is the total number of pixels in block D .

3.3 Proposed Block-Matching Measures

Widely used as a BM measure in many existing BMA's is the SAD between the current block and the candidate block. The SAD measure generally pro-

vides good matching precision, but it is computationally intensive. To minimize the number of computations required for motion estimation, we propose in this chapter to make use of the integral frame attributes to perform the block matching. Specifically, each block under consideration is first partitioned into a number of sub-blocks arranged in a fixed pattern, and the integral frame attributes are then used to compute the feature of each sub-block as a BM parameter. The sub-block patterns that we have examined are shown in Fig. 3.3.

Figure 3.3: Example block patterns: a) 1-block, (b) 2-block, (c) 4-strip, (d) 4-block, (e) 8-strip, (f) 8-block, (g) 16-strip, (h) 16-block, and (i) SAD.

In the following, we examine three potential BM measures by using various block features and integral frame attributes.

3.3.1 SAD of Block Sums (SAD-BS)

This measure uses the $\mathcal{BS}$ as a BM parameter. To locate the best-matching block in the reference frame, the $\mathcal{BS}$'s of all sub-blocks in the current block are compared with those in each candidate block. Specifically, the sum-of-absolute differences between the corresponding $\mathcal{BS}$'s is computed as the BM measure

Figure 3.4: Motion vector of a current block can be obtained by matching the proposed BM measure with that of each candidate block within a search window in the reference frame.

(see Fig. 3.4), given by

$$\text{SAD-BS} = \sum_k |\mathcal{BS}_c(S_k) - \mathcal{BS}_r(S_k)| \quad (3.8)$$

where $\mathcal{BS}_c(S_k)$ and $\mathcal{BS}_r(S_k)$ denote the block sums of the k -th sub-blocks from the current and the candidate blocks, respectively, and can be readily computed by using (3.5).

3.3.2 SAD of Block Variances (SAD-VR)

This measure computes the sum of absolute variance differences from the corresponding sub-blocks as the BM measure, given by

$$\text{SAD-VR} = \sum_k |\delta_c^2(S_k) - \delta_r^2(S_k)| \quad (3.9)$$

where $\delta_c^2(S_k)$ and $\delta_r^2(S_k)$ denote the variances of the k -th sub-blocks from the current and the candidate blocks, respectively, and can be computed by using (3.7).

3.3.3 SAD of Block Sums and Variances (SAD-SV)

This is a hybrid measure combining the above two BM measures. It consists of two parts: the sum of absolute block sum differences and the sum of absolute variance differences from the corresponding sub-blocks, defined as

$$\text{SAD-SV} = \sum_k |\mathcal{BS}_c(S_k) - \mathcal{BS}_r(S_k)| + \lambda N_S \sum_k |\delta_c^2(S_k) - \delta_r^2(S_k)| \quad (3.10)$$

where λ is a proper weighting factor and N_S is the total number of pixels in each sub-block. By verification with a large number of test sequences using different values of λ , we note that when λ is set to 0.02, the proposed SAD-SV measure generally provides the best compression performance. Hence, in our implementation of the proposed SAD-SV measure, we set λ to 0.02.

Conceivably, the performance of the proposed BMA would likely depend on how a block is partitioned into sub-blocks. In general, partitioning a block into too many sub-blocks would increase the computational cost, while too few sub-blocks would decrease the matching precision due to the lack of sufficient spatial granularity. An example of this is illustrated in Fig. 3.5, where the peak-signal-to-noise ratio (PSNR) values obtained by compressing the Foreman test sequence using the FS algorithm together with different proposed BM measures are plotted against the different block patterns shown in Fig. 3.3. The experiment was conducted using the Test Model 5 (TM5) MPEG-2 encoder

Figure 3.5: Performance comparison for the Foreman sequence obtained by using the proposed BM measures computed with different block patterns shown in Fig. 3.3 in conjunction with the FS algorithm.

provided by the MPEG Software Simulation Group at MPEG.org [136] and the target bit-rate, frame rate, and search window size were set to 1.5 Mbits/s, 30 frames/s, and $W = 15$, respectively. The group-of-pictures of the encoded sequence consists of one intra-coded (I) frame followed by nine predictive-coded (P) frames.

3.3.4 Computational Complexity

Consider a video frame of $M \times N$ pixels, a block size of 16×16 pixels, and a search window size of $\pm W$ pixels for block-matching motion estimation. Assume a block pattern of K equal sub-blocks, partitioned into P rows and Q columns, to be used for evaluating the BM measure (see Fig. 3.6 for illustration). To

Figure 3.6: It takes only $(2P + 1) \times Q$ arithmetic operations to compute the proposed SAD-BS measure with a K -block pattern.

compute the proposed SAD-BS measure, we need to calculate the K $\mathcal{BS}_f$'s in the current block and in each of its candidate blocks. As discussed in Section 3.2, 3 arithmetic operations are required to compute the $\mathcal{BS}_f$ of a sub-block. Hence, $3 \times K$ operations are required to calculate the K $\mathcal{BS}_f$'s of a block using the corresponding integral frame attributes. Moreover, we can further reduce the number of operations by exploiting the adjacent property of the sub-blocks as follows.

Consider the P sub-blocks in the i -th column ($1 \leq i \leq Q$) and denote them as $\Psi_i = \{S_{(j-1) \times Q+i} : 1 \leq j \leq P\}$, where the sub-block $S_{(j-1) \times Q+i}$ has the support of $\Omega_{S_{(j-1) \times Q+i}} = \{(x, y) : c_i < x \leq c_{i+1}, r_j < y \leq r_{j+1}\}$. Consider block A_i with support $\Omega_{A_i} = \{(x, y) : c_i < x \leq c_{i+1}, 0 < y \leq r_1\}$ and the blocks in the set $\Gamma = \{A_i \cup \bigcup_{j=1}^t S_{(j-1) \times Q+i} : 1 \leq t \leq P\}$ with support $\Omega_{A_i \cup \bigcup_{j=1}^t S_{(j-1) \times Q+i}} = \{(x, y) : c_i < x \leq c_{i+1}, 0 < y \leq r_{t+1}\}$. We first obtain the $\mathcal{BS}_f$'s of the blocks in

the set Γ from the integral frame attribute $\mathcal{I}_f$ with $P + 1$ subtractions:

$$\begin{aligned} \mathcal{BS}_f(A_i) &= \mathcal{I}_f(c_{i+1}, r_1) - \mathcal{I}_f(c_i, r_1) \\ \mathcal{BS}_f(A_i) + \sum_{j=1}^t \mathcal{BS}_f(S_{(j-1) \times Q + i}) &= \mathcal{I}_f(c_{i+1}, r_{t+1}) - \mathcal{I}_f(c_i, r_{t+1}) \end{aligned} \quad (3.11)$$

for $1 \leq t \leq P$. The $\mathcal{BS}_f$'s of the sub-blocks of Ψ_i can be computed from the above $(P+1)$ $\mathcal{BS}_f$'s by another P subtractions. Hence, to compute all the $\mathcal{BS}_f$'s of the block under consideration, we need a total of $(2P + 1) \times Q$ subtractions.

In addition, to match the current block and a candidate block, $3 \times K - 1$ arithmetic operations (K subtractions, $K - 1$ additions, and K absolute conversions) are required to compute the sum of absolute differences of K corresponding pairs of $\mathcal{BS}_f$'s. Thus, the number of arithmetic operations required to compute the SAD-BS measure at each search location is equal to

$$\mathcal{C}_{BS} = 5 \times PQ + Q - 1 \quad (3.12)$$

Besides the $\mathcal{BS}_f$'s, we also need to calculate the K sub-block variances in the current block and in each of its candidate blocks in order to compute either the SAD-VR measure or the SAD-SV measure. Assuming 1 division also requires 3 arithmetic operations, by using (3.7), a total of 7 arithmetic operations (due to 1 subtraction, 1 multiplication, and 1 division) are required to compute each sub-block variance using $\mathcal{BS}_f$ and $\mathcal{BS}_{f^2}$. Note that the $\mathcal{BS}_{f^2}$'s of a block can be computed in a similar way as the $\mathcal{BS}_f$'s. Hence, the number of arithmetic operations required to compute the proposed SAD-VR and SAD-SV measures

at each search location can be given by, respectively,

$$\begin{aligned}\mathcal{C}_{\text{VR}} &= 14 \times PQ + 2 \times Q - 1 \\ \mathcal{C}_{\text{SV}} &= 18 \times PQ + 2 \times Q - 2\end{aligned}\tag{3.13}$$

Furthermore, the features of many sub-blocks in the reference frame can be reused to compute the BM measures when the proposed measures are used together with the FS algorithm. Therefore, in conjunction with the FS algorithm, it is more efficient to pre-compute the features of all sub-blocks in the reference frame. To exploit the adjacent property of the sub-blocks, we partition a frame into $N - L + 1$ row strips, in which each row strip contains $M - L + 1$ sub-blocks of $L \times L$ pixels. We separate these sub-blocks into L subsets. The first subset contains M/L adjacent sub-blocks, while every other subset contains $M/L - 1$ adjacent sub-blocks. Note that by exploiting this adjacent property, only $2Z + 1$ operations are required to compute the features of Z adjacent sub-blocks. Hence, the number of operations required to compute the features of $M - L + 1$ sub-blocks in a row strip is $2M - L + 2$. Thus, the total number of operations to compute all $L \times L$ sub-block features by using the integral frame attributes is approximately $2MN$.

It should be noted that when block-matching is performed directly on the input video sequence, another $2MN$ and $5MN$ arithmetic operations (see (3.3)) are required to compute each integral frame attribute $\mathcal{I}_f$ and $\mathcal{I}_{f^2}$, respectively. Hence, the total computation overhead of all sub-block sums and sub-block variances are approximately $4MN$ and $18MN$, respectively. Due to the pre-computation of the features of all sub-blocks, the number of operations required to compute the three proposed BM measures at each search location in conjunc-

Table 3.1: Numbers of arithmetic operations required to compute the proposed BM measures at each search location using different block patterns in conjunction with the FS algorithm or a fast search algorithm.

Pattern	Block partition		No. of operations to compute BM measures					
	P	Q	$\mathcal{C}_{\text{FS-BS}}$	$\mathcal{C}_{\text{FS-VR}}$	$\mathcal{C}_{\text{FS-SV}}$	$\mathcal{C}_{\text{BS}}$	$\mathcal{C}_{\text{VR}}$	$\mathcal{C}_{\text{SV}}$
1-block	1	1	2	2	4	5	15	18
2-block	2	1	5	5	10	10	29	36
4-strip	4	1	11	11	22	20	57	72
4-block	2	2	11	11	22	21	59	74
8-strip	8	1	23	23	46	40	113	144
8-block	4	2	23	23	46	41	115	146
16-strip	16	1	47	47	94	80	225	288
16-block	4	4	47	47	94	83	231	294

tion with the FS algorithm is given by, respectively,

$$\begin{aligned}
 \mathcal{C}_{\text{FS-BS}} &= 3 \times PQ - 1 \\
 \mathcal{C}_{\text{FS-VR}} &= 3 \times PQ - 1 \\
 \mathcal{C}_{\text{FS-SV}} &= 6 \times PQ - 2
 \end{aligned} \tag{3.14}$$

Table 3.1 lists the numbers of arithmetic operations required to compute the proposed BM measures at each search location by using different block patterns as shown in Fig. 3.3 in conjunction with the FS algorithm or a fast search algorithm.

On the other hand, when the existing SAD measure is used to match two blocks, at each search location 16×16 pixel pairs are to be compared, and each comparison requires 3 operations—a subtraction, an addition, and an absolute conversion. Table 3.2 shows the total number of operations required per frame when each of the proposed BM measures or the existing SAD measure is used for motion estimation in conjunction with the popular FS and TSS algorithms.

Table 3.2: Numbers of arithmetic operations required by the proposed BM measures and the conventional SAD measure when used together with the FS and TSS algorithms, respectively.

Algorithm	BM measure	No. of arithmetic operations per frame
FS	SAD	$3MN \times (2W + 1)^2$
	SAD-BS	$4MN + \mathcal{C}_{\text{FS-BS}} \times (2W + 1)^2 \times MN/16^2$
	SAD-VR	$18MN + \mathcal{C}_{\text{FS-VR}} \times (2W + 1)^2 \times MN/16^2$
	SAD-SV	$18MN + \mathcal{C}_{\text{FS-SV}} \times (2W + 1)^2 \times MN/16^2$
TSS	SAD	$3MN \times (1 + 8 \times \lceil \log_2 W \rceil)$
	SAD-BS	$2MN + \mathcal{C}_{\text{BS}} \times (1 + 8 \times \lceil \log_2 W \rceil) \times MN/16^2$
	SAD-VR	$7MN + \mathcal{C}_{\text{VR}} \times (1 + 8 \times \lceil \log_2 W \rceil) \times MN/16^2$
	SAD-SV	$7MN + \mathcal{C}_{\text{SV}} \times (1 + 8 \times \lceil \log_2 W \rceil) \times MN/16^2$

Note: There are $(M \times N)/16^2$ blocks per frame.

For a search window with size of $\pm W$ pixels, the number of locations to be searched by the FS algorithm for each block is equal to $(2W + 1)^2$, and that by the TSS algorithm is $1 + 8 \times \lceil \log_2 W \rceil$, where $\lceil \cdot \rceil$ denotes the ceil operator. Hence, in comparison with the SAD measure, the number of the arithmetic operations required by the proposed BM measures can be reduced by factors of

$$\text{Gain}_{\text{FS}} = \frac{3 \times (2W + 1)^2}{\mathcal{R}_{\text{FS}} + \mathcal{C}_{\text{FS}} \times (2W + 1)^2 / 16^2}$$

and

$$\text{Gain}_{\text{TSS}} = \frac{3 \times (1 + 8 \times \lceil \log_2 W \rceil)}{\mathcal{R} + \mathcal{C} \times (1 + 8 \times \lceil \log_2 W \rceil) / 16^2}$$

using the FS and TSS algorithms, respectively, where $\mathcal{C}_{\text{FS}}$ and $\mathcal{C}$, as listed in Table 3.1, are the numbers of operations required to compute the proposed BM measures at each search location by using different block patterns. The constant $\mathcal{R}_{\text{FS}}$ is equal to 4 for the SAD-BS measure and equal to 18 for either SAD-VR or SAD-SV measure, respectively, while the constant $\mathcal{R}$ is equal to 2 for the SAD-BS measure and equal to 7 for either SAD-VR or SAD-SV measure,

Figure 3.7: The gain obtained by different proposed BM measures in comparison with the FS algorithm using the conventional SAD measure.

respectively.

Table 3.3 compares the computational complexity of the proposed BM measures with the use of different block patterns and the existing SAD measure in conjunction with the FS and TSS algorithms. In addition, Fig. 3.7 plots the gain obtained by different proposed BM measures compared with the conventional SAD measure using the FS algorithm.

It should be noted that in comparison with the fast computation schemes (FCS) of the block sum proposed in [49, 50], the same number of arithmetic operations, $4MN$, is required for the pre-computation of all sub-block sums by using the integral frame attributes. Thus, with pre-computation of all sub-block sums the total number of operations required per frame for motion estimation using a block pattern of K equal sub-blocks is approximately the same for either

Table 3.3: Numbers of arithmetic operations required by the proposed BM measures using different block patterns and the complexity comparison against the FS and TSS algorithms using the conventional SAD measure for a case study with $W = 15$, $M = 352$, and $N = 288$.

Patterns	SAD-BS		SAD-VR		SAD-SV	
	No. of ops	Percentage	No. of ops	Percentage	No. of ops	Percentage
FS	292,267,008	100.00%	292,267,008	100.00%	292,267,008	100.00%
1-block	1,166,616	0.40%	2,180,376	0.75%	2,941,488	1.01%
2-block	2,308,284	0.79%	3,322,044	1.14%	5,224,824	1.79%
4-strip	4,591,620	1.57%	5,605,380	1.92%	9,791,496	3.35%
4-block	4,591,620	1.57%	5,605,380	1.92%	9,791,496	3.35%
8-strip	9,158,292	3.13%	10,172,052	3.48%	18,924,840	6.48%
8-block	9,158,292	3.13%	10,172,052	3.48%	18,924,840	6.48%
16-strip	18,291,636	6.26%	19,305,396	6.61%	37,191,528	12.73%
16-block	18,291,636	6.26%	19,305,396	6.61%	37,191,528	12.73%

a) In comparison with the FS algorithm using the SAD measure.

Patterns	SAD-BS		SAD-VR		SAD-SV	
	No. of ops	Percentage	No. of ops	Percentage	No. of ops	Percentage
TSS	10,036,224	100.00%	10,036,224	100.00%	10,036,224	100.00%
1-block	268,092	2.67%	905,652	9.02%	944,856	9.41%
2-block	333,432	3.32%	1,088,604	10.85%	1,180,080	11.76%
4-strip	464,112	4.62%	1,454,508	14.49%	1,650,528	16.45%
4-block	477,180	4.75%	1,480,644	14.75%	1,676,664	16.71%
8-strip	725,472	7.23%	2,186,316	21.78%	2,591,424	25.82%
8-block	738,540	7.36%	2,212,452	22.04%	2,617,560	26.08%
16-strip	1,248,192	12.44%	3,649,932	36.37%	4,473,216	44.57%
16-block	1,287,396	12.83%	3,728,340	37.15%	4,551,624	45.35%

b) In comparison with the TSS algorithm using the SAD measure.

the FCS scheme or the proposed scheme and given by

$$\mathcal{T}_1 = 4MN + \text{No. of search points} \times (3K - 1) \times MN/16^2 \quad (3.15)$$

When the number of locations to be searched is small, pre-computation of all sub-block sums is unnecessary and inefficient. The total number of operations required per frame for motion estimation by using the proposed scheme without pre-computation of all sub-block sums is given by

$$\mathcal{T}_2 = 2MN + \text{No. of search points} \times C_{\text{BS}} \times MN/16^2 \quad (3.16)$$

where C_{BS} for different block patterns is listed in Table 3.1. It is easy to see that when the number of search points or the number of sub-blocks in the block pattern used is small, pre-computing all sub-block sums may be not necessary and efficient (i.e., $\mathcal{T}_2 < \mathcal{T}_1$ due to the less computation overhead of $2MN$ arithmetic operations). Thus, in such cases the proposed scheme is more efficient than the FCS scheme since the BM measure can be evaluated without pre-computing all sub-block sums.

Table 3.4 shows the number of operations required per frame by using the existing FCS scheme and the proposed scheme without pre-computing all sub-block sums for evaluating the SAD-BS measure in conjunction with three fast search algorithms, namely the TSS, gradient descent search (GDS) [34], and motion vector field adaptive search technique (MVFAST) [41]. The results show that the proposed scheme using the TSS algorithm without pre-computing all sub-block sums can achieve a lower computational complexity for most of the block patterns with few sub-blocks, including the preferred 4-block pattern

Table 3.4: The numbers of operations required by using the fast computation scheme (FCS) of the block sum and the proposed scheme without pre-computing all sub-block sums for evaluating the SAD-BS measure in conjunction with the TSS, the gradient descent search (GDS), and motion vector field adaptive search technique (MVFAST).

Pattern	TSS (Average no. of search points = 33)		GDS (Average no. of search points = 9)		MVFAST (Average no. of search points = 6)	
	Proposed		Proposed		Proposed	
	FCS	scheme	FCS	scheme	FCS	scheme
1-block	431,640	268,092	412,632	220,572	410,256	214,632
2-block	470,844	333,432	423,324	238,392	417,384	226,512
4-strip	549,252	464,112	444,708	274,032	431,640	250,272
4-block	549,252	477,180	444,708	277,596	431,640	252,648
8-strip	706,068	752,472	487,476	345,312	460,152	297,792
8-block	706,068	738,540	487,476	348,876	460,152	300,168
16-strip	1,019,700	1,248,192	573,012	487,872	517,176	392,832
16-block	1,019,700	1,287,396	573,012	498,564	517,176	399,960

(see Section 3.5.1). Furthermore, for block patterns with many sub-blocks and resulting in $\mathcal{T}_2 > \mathcal{T}_1$ such as the 16-block pattern when used together with the TSS algorithm, we can pre-compute all sub-block features like in the case of the FS algorithm and achieve virtually the same computational complexity as the FCS scheme.

In addition, it can be seen from (3.15) and (3.16), the smaller the number of search points or the number of sub-blocks in the block pattern used, the lower the computational complexity is required for the proposed scheme without pre-computing all sub-block sums as compared to the FCS scheme. This is reconfirmed in the results shown in Table 3.4, where our proposed scheme can have a much lower computational complexity than the FCS scheme when used together with a fast search algorithm whose number of search points is small such as the GDS or MVFAST algorithm. This shows that when the number

of search points is small, pre-computation of all sub-block sums is unnecessary and inefficient. By a simple calculation, it can be seen that when the average number of search points is less than 15, which is the case for most recent fast search algorithms such as GDS, MVFAST, and cross-diamond search (CDS), the proposed scheme always outperforms the FCS scheme for any block pattern used. Furthermore, the proposed scheme can be easily extended to other block features (e.g., block variance) and a two-step approach described in the next section.

In short, our scheme can always achieve either approximately the same complexity by pre-computing all sub-block sums or lower complexity when used together with fast search algorithms having small number of search points or block patterns with few sub-blocks.

3.4 Extended Use of Proposed BM Measures

3.4.1 Two-Step Approach

We have shown that by partitioning a block into many sub-blocks and using integral frame attributes, the number of computations required to evaluate the proposed BM measures can be notably reduced. However, there is a trade-off between the performance and the computational complexity when the number of sub-blocks increases. For example, the 16-block pattern can achieve a compression performance close to that of the existing SAD measure, but it is more computationally intensive compared with the 1-block pattern (see Figs. 3.5 and 3.7). We shall refer to a block pattern with few sub-blocks (such as the 1-block or 2-block pattern) as a coarse block pattern, and that with many sub-blocks

(such as 16-strip or 16-block pattern) as a fine block pattern. In general, a coarse block pattern can provide a low matching accuracy but a high reduction in the computational cost. The reverse is true for a fine block pattern.

Exploiting this property and the concept of hierarchical approach (refer to Chapter 2, Section 2.3.1), we propose to integrate the proposed BM measures into a two-step approach using different block patterns in each step. The basic idea here is to compensate for the lack of discriminating power of the coarse block pattern by using the fine block pattern with the increased spatial granularity in order to reduce the uncertainty. Specifically, we use a coarse block pattern in the first step to gain high computation savings, and a fine block pattern in the second step to attain a high matching accuracy. The proposed two-step approach is summarized as follows:

Step 1: A coarse block pattern is used in the proposed BM measure and an exhaustive search is performed in a search window with the size of $W \times W$. The K candidates that give the smallest values of the BM measure are identified for further refinement in Step 2. These are referred to as good candidate blocks as the best-matching block is likely to be one of them.

Step 2: The K good candidate blocks are re-evaluated by using the proposed BM measure with a fine block pattern in order to find the best-matching block.

Clearly, the more the number of good candidate blocks K identified in the first step for further refinement, the closer the performance is to that obtained by using only the fine block pattern in the originally proposed BMA (referred to as the one-step approach). However, the larger the value of K , the higher the computational complexity is.

To illustrate the aforementioned property, let $\Gamma = \{(mv_{x_i}, mv_{y_i}) : 1 \leq i \leq$

$K\}$ be the motion vectors of the K good candidate blocks identified in Step 1. Let (mv_x, mv_y) be the motion vector obtained by using the same fine block pattern in the one-step approach. Define the minimum Euclidean distance between (mv_x, mv_y) and Γ as

$$d_{\min} = \min \left\{ \sqrt{(mv_x - mv_{x_i})^2 + (mv_y - mv_{y_i})^2} \right\} \quad (3.17)$$

for $1 \leq i \leq K$. We say (mv_x, mv_y) is within 1-pixel neighborhood of the set of motion vectors Γ if and only if $d_{\min} \leq \sqrt{2}$, and define the hit ratio as the probability that the motion vector obtained by using the one-step approach (using the same fine block pattern) is within 1-pixel neighborhood of the motion vectors of the K good candidate blocks identified in Step 1 (using a coarse block pattern) of the two-step approach.

Fig. 3.8 shows the hit ratio as a function of K obtained empirically over a large number of video test sequences for two cases: 1) 1-block and 16-block patterns are used in Step 1 and Step 2, respectively, 2) 2-block and 16-block patterns are used in Step 1 and Step 2, respectively. Obviously, the more number of good candidate blocks identified in Step 1, the higher the hit ratio is, and the better the compression performance is achieved. However, it is also evident that when the hit ratio is greater than 95%, it increases slowly with the increase of the value of K . Hence, identifying the number of good candidate blocks to obtain a hit ratio around 95% could provide a good trade-off between the compression performance and computational complexity. Such a good trade-off can be obtained by setting the values of K to 90 and 50, respectively, for the two cases shown in Fig. 3.8. Our study shows that by retaining and refining only such a small number of good candidate blocks, the proposed two-step approach

Figure 3.8: Empirical hit ratios for different values of K in two cases: 1) Step1: 1-block pattern, Step 2: 16-block pattern, 2) Step 1: 2-block pattern, Step 2: 16-block pattern.

can perform marginally close to the one-step approach using the same fine block pattern in Step 2, but incurring a significant less computational complexity.

Computational Complexity Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be the numbers of operations required to use the coarse and fine block patterns, respectively, in the two-step approach for computing the proposed BM measure at each location (see Table 3.1). The number of operations required to find the best-matching block for each block under consideration can be given by

$$\mathcal{C}_1 \times (2W + 1)^2 + \mathcal{C}_2 \times K \quad (3.18)$$

Table 3.5 shows the total number of operations required per frame when the

Table 3.5: Numbers of arithmetic operations required by the proposed BM measures in the proposed two-step approach and by the conventional SAD measure when used together with the FS algorithm.

BM measure	No. of arithmetic operations per frame
SAD	$3MN \times (2W + 1)^2$
SAD-BS	$6MN + [\mathcal{C}_{\text{FS-BS1}} \times (2W + 1)^2 + \mathcal{C}_{\text{FS-BS2}} \times K] \times MN/16^2$
SAD-VR	$29MN + [\mathcal{C}_{\text{FS-VR1}} \times (2W + 1)^2 + \mathcal{C}_{\text{FS-VR2}} \times K] \times MN/16^2$
SAD-SV	$29MN + [\mathcal{C}_{\text{FS-SV1}} \times (2W + 1)^2 + \mathcal{C}_{\text{FS-SV2}} \times K] \times MN/16^2$

proposed two-step approach and the existing SAD measure are used for motion estimation in conjunction with the FS algorithm. In comparison with the SAD measure, the number of operations required by the proposed two-step approach can be reduced by a factor of

$$\text{Gain}_{\text{FS,2-step}} = \frac{3 \times (2W + 1)^2}{\mathcal{R} + [\mathcal{C}_1 \times (2W + 1)^2 + \mathcal{C}_2 \times K] / 16^2}$$

where the constant $\mathcal{R}$ is equal to 6 for the SAD-BS measure and equal to 29 for either SAD-VR or SAD-SV measure.

To illustrate the amount of reduction, let us consider two combinations of options in the proposed two-step approach: 1) Step 1: 1-block pattern, Step 2: 16-block pattern, $K = 90$, 2) Step 1: 2-block pattern, Step 2: 16-block pattern, $K = 50$. Using a search window with $W = 15$, the number of operations can be reduced by factors of 96 and 85, respectively, while still attaining compression performance that is marginally close to that by using the 16-block pattern in the one-step approach. It should also be noted that the sub-block features for different block patterns can be computed with only one-time computation overhead for the integral frame attributes to minimize the complexity.

3.4.2 Partial Summation Elimination

Apart from the two-step approach, we have also made use of the proposed BM measures in conjunction with a partial summation elimination (PSE) scheme to further speed up the motion estimation while maintaining the same compression performance. In this section, we shall consider only the SAD of the sub-block features as the BM measure, and refer to it as the complete matching error because it is accumulated over the whole block under consideration and defined as

$$\text{SAD-BF} = \sum_{p=1}^K |\text{BF}_c(\text{S}_p) - \text{BF}_r(\text{S}_p)| \quad (3.19)$$

where $\text{BF}_c(\text{S}_p)$ and $\text{BF}_r(\text{S}_p)$ denote the features of the p -th sub-blocks from the current and the candidate blocks, respectively, and K is the total number of sub-blocks in the block pattern used.

As the matching error of the best-matching candidate block is much smaller than that of most other candidate blocks, summing up the absolute feature differences of all the sub-blocks to obtain the complete matching error for block matching may be not necessary for all the candidate blocks. Instead, partial sum of the matching error is able to eliminate many impossible candidate blocks, thus saving the number of computations. Specifically, we define the k -th partial sum of sub-block feature differences (PSAD-BF) as

$$\text{PSAD-BF}^k = \sum_{p=1}^k |\text{BF}_c(\text{S}_p) - \text{BF}_r(\text{S}_p)| \quad (3.20)$$

Let $\text{SAD-BF}_{\text{bsf}}$ denote the best-so-far matching error. The main idea of PSE is that a candidate block can be rejected immediately and the remaining compu-

Figure 3.9: Flowchart of the proposed PSE scheme.

tation of the BM measure is unnecessary if there exists a value of k smaller than K , whose PSAD-BF^k already exceeds the best-so-far matching error. Fig. 3.9 shows the flowchart of the PSE scheme, in which the absolute differences of the sub-block features are summed up from the top to the bottom of each column, and from the left column to the right column.

It should be noted that the efficiency of using a PSE scheme to reject impossible candidate blocks depends on how well the search algorithm allows a

close-to-optimal matching error to be detected in the early stage of the search. To meet this objective, we employ a spiral search algorithm: the search begins at the original checking point, point $(0, 0)$ in the search window, and then moves outward in a spiral-scanning path. This search algorithm is to exploit the center-biased property of the distribution of motion vectors in many real-world video sequences. However, this center-biased property may not hold when there is a dominant object or camera motion. To save more computations for this type of videos, we thus exploit the spatial correlation property of the motion vectors to determine a good initial search point that results in a low matching error in the early stage of the search. Specifically, we use the motion vectors of two adjacent blocks in the current frame (left and top) and a zero motion vector to predict the motion vector of the current block. We compute the matching errors corresponding to these three motion vectors and select the one that gives the smallest matching error as the initial search point (the center) of the spiral search.

We illustrate in Fig. 3.10 a real-case example to show that how a good prediction of the initial search point can reduce the number of computations. It can be observed that using a PSE scheme with the prediction of initial search point the number of computations can be reduced notably as the matching error obtained in the early search step is much smaller than that without using a prediction of the search center.

(a) Using the origin, point $(0, 0)$ of the search window, as the initial search center.

(b) Using the point predicted by the proposed method as the initial search center.

Figure 3.10: The number of sub-blocks used for computing PSAD-BF both with and without the prediction of the initial search center for each candidate block in the search window.

3.5 Experimental Results

We have conducted a series of experiments to evaluate the performance of the proposed BMA. Our test sequences include ten popular CIF resolution (352×288) sequences: Coastguard (300), Container (300), Flower (250), Football (125), Foreman (300), Mother & Daughter (270), News (300), Stefan (300), Tempete (258), and Tennis (370), where the total frame number of each sequence is indicated in the parenthesis. These sequences contain different amounts of motion and spatial details, and have been widely tested in the research community of video compression.

We conducted the experiments by using the Test Model 5 (TM5) MPEG-2 encoder provided by the MPEG Software Simulation Group at MPEG.org [136]. For each test sequence, we set the target bit-rate, frame rate, and search window size to 1.5 Mbits/s, 30 frames/s, and $W = 15$, respectively.

To evaluate the compression performance of our proposed BM measures by using the different block patterns as shown in Fig. 3.3, we encoded ten test sequences in conjunction with both the FS and TSS algorithms. The group-of-pictures (GOP) of each encoded sequence consists of one intra-coded (I) frame followed by nine predictive-coded (P) frames.

3.5.1 Proposed BM Measures

In Fig. 3.11, we show the average PSNR results for the Coastguard sequence obtained by using the three proposed BM measures in conjunction with both the FS and TSS algorithms. The results show that, in comparison with the existing SAD measure, our BM measures can reduce a significant computation load in

(a) In conjunction with the FS algorithm.

(b) In conjunction with the TSS algorithm.

Figure 3.11: Performance comparison obtained by using the proposed BM measures computed with different block patterns shown in Fig. 3.3 for the Coast-guard sequence.

Table 3.6: The PSNR results (in dB) of ten test sequences obtained by using the proposed SAD-BS measure with various block patterns and the conventional SAD measure using the FS algorithm. (The average PSNR difference is computed against that of the SAD measure.)

Sequence	1-block	2-block	4-strip	4-block	8-strip	8-block	16-strip	16-block	SAD
Coastguard	30.7	31.8	33.6	33.6	34.0	34.2	34.1	34.5	34.5
Container	34.5	36.9	38.6	38.6	38.8	38.8	38.8	38.8	38.8
Flower	24.5	26.0	28.6	28.7	29.0	29.2	29.0	29.3	29.4
Football	27.2	27.7	28.7	28.9	29.0	29.3	29.1	29.4	29.5
Foreman	33.9	35.2	36.3	36.4	36.6	36.8	36.7	37.0	37.1
M & D	38.7	40.0	40.9	41.0	41.0	41.1	41.0	41.1	41.1
News	37.5	40.5	41.3	41.4	41.4	41.5	41.4	41.5	41.5
Stefan	27.4	28.6	30.6	30.9	31.1	31.5	31.2	31.7	31.8
Tempete	28.0	29.2	30.8	31.0	31.1	31.3	31.2	31.4	31.5
Tennis	27.3	28.8	30.0	30.0	30.3	30.4	30.4	30.5	30.6
Average PSNR diff.	3.45	1.97	0.60	0.49	0.34	0.16	0.27	0.06	0.00

motion estimation without degrading much the compression performance. It is noted that, although being more computationally intensive, the SAD-VR measure cannot provide a better compression performance as compared with the SAD-BS measure. However, when block variances are combined with block sums with a proper weighting factor, the SAD-SV measure is able to perform better than SAD-BS. The computational complexities of the SAD-VR and the SAD-SV measures are much higher than that of the SAD-BS measure.

Tables 3.6 and 3.7 show the PSNR results of the ten test sequences using the SAD-BS, SAD-SV, and the existing SAD measures in conjunction with the FS algorithm. Not surprisingly, the performance of the proposed BM measures improves as the number of the sub-blocks increases (i.e., the size of each sub-block decreases), for the reason that the video spatial variations can be gauged more closely when block matching is evaluated based on means and variances

Table 3.7: The PSNR results (in dB) of ten test sequences obtained by using the proposed SAD-SV measure with various block patterns and the conventional SAD measure using the FS algorithm. (The average PSNR difference is computed against that of the SAD measure.)

Sequence	1-block	2-block	4-strip	4-block	8-strip	8-block	16-strip	16-block	SAD
Coastguard	31.8	33.2	33.9	34.0	34.1	34.3	34.2	34.5	34.5
Container	36.9	38.3	38.7	38.8	38.8	38.8	38.8	38.8	38.8
Flower	25.7	28.1	28.9	29.1	29.1	29.3	29.1	29.3	29.4
Football	27.7	28.6	29.0	29.2	29.2	29.4	29.2	29.5	29.5
Foreman	35.1	36.2	36.6	36.7	36.8	36.9	36.8	37.0	37.1
M & D	39.8	40.8	41.0	41.0	41.1	41.1	41.1	41.1	41.1
News	40.4	41.3	41.4	41.4	41.4	41.5	41.4	41.5	41.5
Stefan	28.6	30.4	31.1	31.4	31.3	31.6	31.4	31.8	31.8
Tempete	29.2	30.7	31.1	31.3	31.2	31.4	31.3	31.4	31.5
Tennis	28.5	29.7	30.2	30.2	30.4	30.5	30.4	30.5	30.6
Average PSNR diff.	2.06	0.79	0.36	0.25	0.23	0.10	0.20	0.05	0.00

in smaller sub-blocks. However, it is also evident that only a marginal performance gain can be obtained by using block patterns with sizes of sub-blocks smaller than that of the 4-block pattern. In other words, the 4-block pattern can generally provide a good trade-off between the computational complexity and compression performance. Furthermore, using the 4-block pattern and incurring a much lower computational cost, the SAD-BS measure can achieve a compression performance comparable with that of the existing SAD measure. In particular, with a computational complexity reduced by a factor of 64, the proposed SAD-BS measure can obtain average PSNR results only about 0.5 dB inferior to that obtained by the existing SAD measure. Furthermore, we note that the block patterns shown in Fig. 3.3 can generally perform better than the strip patterns having the same size, while incurring almost the same number of computations. This performance difference can generally be attributed to

Table 3.8: The PSNR results (in dB) of ten test sequences obtained by using the proposed SAD-BS measure with various block patterns and the conventional SAD measure using the TSS algorithm. (The average PSNR difference is computed against that of the SAD measure.)

Sequence	1-block	2-block	4-strip	4-block	8-strip	8-block	16-strip	16-block	SAD
Coastguard	31.2	32.1	33.5	33.4	33.8	34.1	34.0	34.3	34.3
Container	36.7	38.2	38.7	38.7	38.8	38.8	38.8	38.8	38.8
Flower	24.9	25.8	28.1	27.8	28.7	28.9	28.8	29.0	28.8
Football	27.5	28.1	28.6	28.8	28.8	29.1	28.8	29.3	29.2
Foreman	34.5	35.4	36.0	36.1	36.1	36.4	36.2	36.6	36.6
M & D	39.8	40.6	40.9	41.0	41.0	41.1	41.0	41.1	41.1
News	40.2	41.1	41.3	41.3	41.3	41.4	41.3	41.5	41.5
Stefan	27.9	28.8	30.3	30.3	30.7	31.1	30.8	31.0	30.9
Tempete	29.2	30.3	31.0	31.1	31.2	31.3	31.2	31.4	31.5
Tennis	28.6	29.5	30.0	30.1	30.2	30.3	30.2	30.4	30.3
Average PSNR diff.	2.09	1.20	0.42	0.40	0.22	0.05	0.17	-0.04	0.00

the fact that the block patterns enclose the same number of pixels in the horizontal and vertical directions and can estimate the horizontal motions, which are common in the test sequences, more accurately as compared with the strip patterns.

Table 3.8 shows the PSNR results of the ten test sequences using the SAD-BS measure and the existing SAD measure in conjunction with the TSS algorithm. Similar to the case of the FS algorithm, the 4-block pattern together with the TSS algorithm can achieve a compression performance only about 0.4 dB inferior to that obtained by the existing SAD measure, while incurring a computational complexity reduced by a factor of 21. It should also be noted that when applying the proposed measures in other fast search algorithms with a small number of search points such as GDS and MVFAST, the average PSNR degradation is only about 0.3 dB for the preferred 4-block pattern when compared to the SAD

measure. The results may shed light on the fact that our proposed BM measures can also be applied together with a fast BMA to reduce the computational complexity without degrading much the video quality.

To evaluate the performance of the proposed BM measures across a range of bit rates, we have conducted the experiments to encode the test sequences using the proposed BM measures at different target bit rates. The preferred 4-block pattern together with the proposed SAD-BS measure were used in conjunction with the FS and TSS algorithms. Figs. 3.12 and 3.13 show the rate-distortion curves obtained by the proposed BM measure and the conventional SAD measure for the Foreman and Tempete sequences, respectively. The results show that the performance of the proposed BM measures is consistent over a range of bit rates.

To see that the PSNR differences are also uniformly distributed over each entire sequence, the frame-to-frame PSNR results of the Foreman sequence obtained by using different BM measures are also shown in Fig. 3.14 for comparison. It should also be noted that such small PSNR differences are usually hard to perceive under normal viewing conditions. This is evident from Fig. 3.15, which shows sample frames of the Foreman sequence obtained by the proposed SAD-BS measure and the existing SAD measure.

3.5.2 Extended Use of Proposed BM Measures

To evaluate the performance of using the proposed BM measures in the two-step approach, we encoded four test sequences—Foreman, Coastguard, Stefan, Flower—by using the proposed SAD-BS measure in conjunction with two cases of block patterns in the two steps: 1) Step 1: 1-block pattern, Step 2: 16-block

(a) In conjunction with the FS algorithm.

(b) In conjunction with the TSS algorithm.

Figure 3.12: Rate-distortion curves obtained by the proposed SAD-BS measure together with the preferred 4-block pattern and the conventional SAD measure for the Foreman sequence.

(a) In conjunction with the FS algorithm.

(b) In conjunction with the TSS algorithm.

Figure 3.13: Rate-distortion curves obtained by the proposed SAD-BS measure together with the preferred 4-block pattern and the conventional SAD measure for the Tempete sequence.

Figure 3.14: Frame-to-frame PSNR results of the Foreman sequence obtained by the proposed SAD-BS measure and the conventional SAD measure using the FS and TSS algorithms.

pattern, and 2) Step 1: 2-block pattern, Step 2: 16-block pattern. Figs. 3.16 and 3.17 plot the average PSNR performances of these four sequences with various numbers of good candidate blocks selected in Step 1 for the two cases. The results show that by using the fine block pattern with the increased spatial granularity in the two-step approach, it helps to reduce the uncertainty due to the lack of discriminating power of the coarse block pattern and will be able to achieve the performance close to that of using the fine block pattern in the one-step approach. Obviously, the more number of good candidate blocks selected in Step 1, the closer is the performance to that obtained by using the proposed SAD-BS measure applying the 16-block pattern in the one-step approach. It should be noted when the number of good candidate blocks are equal to 90

SAD + FS (37.6 dB)

SAD-BS + FS (37.1 dB)

SAD + TSS (37.3 dB)

SAD-BS + TSS (36.9 dB)

Figure 3.15: Sample frames from the Foreman sequence encoded by using the proposed SAD-BS measure and the existing SAD measure using the FS and TSS algorithms.

Figure 3.16: The PSNR results obtained by the proposed two-step approach and the SAD-BS measure using the 1-block and 16-block patterns in Step 1 and Step 2, respectively, for various number of good candidate blocks retaining in Step 1.

Figure 3.17: The PSNR results obtained by the proposed two-step approach and the SAD-BS measure using the 2-block and 16-block patterns in Step 1 and Step 2, respectively, for various number of good candidate blocks retaining in Step 1.

and 50 for case 1 and case 2, respectively, the proposed two-step approach can achieve similar compression performance as the proposed SAD-BS measure with the use of the 16-block pattern in the one-step approach. And the number of computations can be reduced by factors of 96 and 85, respectively, compared with the existing SAD measure using the FS algorithm. The results

As the PSE scheme proposed in this chapter would not result further compression degradation, the quality of the compressed video obtained by using this scheme is the same as that obtained by the original proposed BMA. For comparison purpose, we performed the experiments by using the proposed BMA both with and without using the PSE scheme. In addition, we also conducted the experiments of the PSE scheme both with and without using the initial search center prediction. Table 3.9 shows the average number of operations required by using different block patterns for two test sequences—Foreman and Stefan. It is easy to see that using the PSE scheme can further reduce the computational complexity of the proposed BMA. Specifically, the number of computations for the 4-block pattern and 16-block pattern can be reduced by factors of 2.87 and 4.38 for the Foreman sequence and by factors of 2.71 and 3.56 for the Stefan sequence, respectively. Furthermore, by using the prediction of initial search center in the PSE scheme, the number of computations for motion estimation can be further reduced.

3.6 Summary

We have presented in this chapter new block-matching measures for reducing the computational complexity of block-based motion estimation. The proposed measures evaluate the match between two blocks based on features such as

Table 3.9: Average number of operations per block taken by the proposed BMA with and without using the PSE scheme for the Foreman and Stefan sequences. PSE₂ and PSE₁ denote the use of the PSE scheme with and without the prediction of initial search center.

(a) Foreman

Algorithm	Absolution	Add/Sub	Comparison	Total	Speed Up
2-block	1,738.67	6,954.67	868.33	9,561.67	1.00
2-block + PSE ₁	932.84	3,731.38	932.84	5,597.07	1.71
2-block + PSE ₂	919.32	3,677.27	919.32	5,515.91	1.73
4-block	3,477.33	14,778.67	868.33	19,124.33	1.00
4-block + PSE ₁	1,159.44	4,704.72	1,159.44	7023.61	2.72
4-block + PSE ₂	1,102.24	4,454.47	1,102.24	6,658.94	2.87
8-block	6,954.67	28,688.00	868.33	36,511.00	1.00
8-block + PSE ₁	1,721.14	6,954.99	1,721.14	10,397.28	3.51
8-block + PSE ₂	1,577.28	6,355.23	1,577.28	9,509.80	3.84
16-block	13,909.33	58,245.33	868.33	73,023.00	1.00
16-block + PSE ₁	3,057.49	12,544.85	3,057.49	18,659.83	3.91
16-block + PSE ₂	2,739.21	11,197.71	2,739.21	16,676.13	4.38

(b) Stefan

Algorithm	Absolution	Add/Sub	Comparison	Total	Speed Up
2-block	1,738.67	6,954.67	868.33	9,561.67	1.00
2-block + PSE ₁	946.93	3,787.72	946.93	5,681.58	1.68
2-block + PSE ₂	932.08	3,728.33	932.08	5,592.50	1.71
4-block	3,477.33	14,778.67	868.33	19,124.33	1.00
4-block + PSE ₁	1,238.25	5,046.80	1,238.25	7,523.31	2.54
4-block + PSE ₂	1,164.26	4,723.35	1,164.26	7,051.87	2.71
8-block	6,954.67	28,688.00	868.33	36,511.00	1.00
8-block + PSE ₁	1,996.37	8,093.32	1,996.37	12,086.06	3.02
8-block + PSE ₂	1,794.14	7,249.05	1,794.14	10,837.33	3.37
16-block	13,909.33	58,245.33	868.33	73,023.00	1.00
16-block + PSE ₁	3,827.41	15,796.89	3,827.41	23,451.72	3.11
16-block + PSE ₂	3,359.11	13,813.64	3,359.11	20,531.87	3.56

block sum and block variance, which can be easily computed from integral frame attributes. Analytical study and experimental results show that, compared with the existing SAD measure, the proposed BM measures can reduce the number of arithmetic operations by factors of 64 and 21, when used in conjunction with the FS and TSS algorithms, respectively. This saving in computation is gained without degrading much the compressed video quality. Furthermore, by using a two-step approach, in which different types of block patterns are used in each step, the number of computations can be further reduced by a factor of 96 when used together with the FS algorithm. In addition, the proposed BM measures can be used together with a partial summation elimination scheme to reject impossible candidate blocks in the early stage of the search, thus reducing the computational complexity while maintaining the same performance of the original proposed BM measures. It should be noted that, besides the two search algorithms examined in this chapter, the proposed measures could also be used in many existing fast search algorithms to reduce the computational complexity.

Chapter 4

Video Transcoding Between H.263 and H.264/AVC Standards

In this chapter, we focus on the optimized techniques for transcoding video between different coding standards to reduce the computational complexity. We consider a transcoding problem between the existing standards and the latest H.264/AVC standard, which has employed many advanced technologies. A fast motion vector re-estimation scheme together with an efficient intra-prediction mode selection are proposed. In addition, we present an enhanced rate control algorithm for H.264/AVC transcoding to obtain a better visual quality for the transcoded video.

This chapter is organized as follows. Section 4.1 provides some background and motivations. Section 4.2 gives a brief analysis of complexity and performance issues in H.264/AVC transcoding. Section 4.3 describes the proposed ef-

efficient methods for syntax transcoding and downsizing transcoding from H.263 to H.264/AVC standard as well as the enhanced rate control method. The experimental results are shown in Section 4.4. In Section 4.5, we conclude the chapter by summarizing the main contributions.

4.1 Introduction

Video contents are generally compressed to meet closely the constraints of the target applications. Thus, videos compressed for one application may not be well suited for other applications subject to a set of more restricted constraints, e.g., a lower channel capacity or a smaller display screen. To a certain extent, this mismatch in application constraints has hindered efficient sharing of compressed videos among today's heterogeneous networks and devices.

To address such inefficiency, video transcoding has played an important role to convert an existing compressed video to a new compressed video in a different format or syntax. Video transcoding techniques can be broadly classified into homogenous and heterogenous transcodings. Homogeneous transcoding is generally used to reduce the bit rate, frame rate and/or spatial resolution (downsizing transcoding) so that the processed video can suit better the new application constraints. On the other hand, heterogenous transcoding is used to change the syntax of a compressed video (syntax transcoding) for decoders compliant to a different compression standard. To meet the requirements of many potential real-time applications, existing video transcoding techniques mostly focus on a few computationally intensive encoding functions (e.g., motion estimation or discrete cosine transform) to speed up the transcoding process. Many

also exploit the information extracted from the precoded video [57, 72–76, 81].

Meanwhile, in response to the need of a more efficient video coding technique for diversified networks and applications, the H.264/AVC video coding standard has been recently developed and standardized collaboratively by the ITU-T VCEG and the ISO/IEC MPEG standard committees. The standard achieves high coding efficiency by employing a number of new technologies, including multiple reference frames, variable block sizes for motion estimation and compensation, intra-prediction coding, 4×4 integer transform, in-loop deblocking filter, etc. Empirical studies have shown that H.264/AVC can achieve up to approximately 50% bit rate savings for similar perceived video quality as compared with other existing standards, such as H.263 and MPEG-4. In view of this much improved performance, it is expected that a large number of videos and devices compliant to the H.264/AVC standard will soon become popular. Hence, there is a need for transcoding precoded videos to H.264/AVC format.

However, due to its new coding features, H.264/AVC is much more different and complex than other existing standards. For example, multiple reference frames and variable block sizes make the motion estimation in H.264/AVC much more complex than that of other standards. Besides motion estimation, intra-prediction and coding mode decision in a rate-distortion optimized fashion also increase the coding complexity substantially. In addition, these new features also make accurate rate control more difficult and challenging for both coding and transcoding in the H.264/AVC standard [137]. Due to these differences, direct application of existing transcoding techniques may not be efficient and suitable for this new standard.

Several works have been conducted to address the transcoding issues be-

tween the existing standards and H.264/AVC by considering these advanced coding features. In [138,139], fast transcoding methods for intra frames between MPEG-2 and H.264/AVC are presented. Specifically, Yu *et al.* [138] exploited the similarity between the directional features of MPEG-2 in frequency domain and spatial-domain prediction in H.264/AVC and used such side information to simplify the intra-prediction mode selection required for the transcoded video. Meanwhile, Kalva *et al.* [139] reduced the complexity by decomposing intra-prediction operation into some sparse matrixes using the property of DCT. In addition, a number of research has analyzed and made use of the inherent relationship between the precoded and transcoded videos to provide efficient mode mapping and motion estimation methods [140–144]. These methods gain the computation savings by using some fast motion estimation and/or mode decision based on the information in the precoded video rather than using the expensive rate-distortion optimization process. Furthermore, to address the mismatch of the transformation between the existing standards and H.264/AVC, several works have been conducted to efficiently convert the DCT into integer transform coefficients in the compressed domain [145,146].

In this chapter, we investigate and propose efficient methods for transcoding videos compressed in H.263 format to H.264/AVC standard by exploiting the new coding features. Specifically, the proposed methods aim to reduce the computational complexity while maintaining acceptable video quality for syntax transcoding and 2:1 downsizing transcoding from H.263 to H.264/AVC standard. In a nutshell, the proposed methods include three components, namely fast intra-prediction mode selection, motion vector re-estimation and inter mode selection, and enhanced rate control for H.264/AVC transcoding. The first two components focus on the most computationally intensive parts of

the H.264/AVC standard to speed up the transcoding process, while the third component aims to achieve a better video quality by enhancing the H.264/AVC rate control with the side information extracted from the precoded video. The experimental results show that the proposed methods can reduce the total encoding time by a factor of 6 and suffer only about 0.35-dB loss in the peak-signal-to-noise ratio (PSNR).

4.2 Complexity and Performance Issues in H.264/AVC Transcoding

The H.264/AVC standard incorporates a set of new coding features to achieve its high coding efficiency at the cost of substantial increase in complexity. In this section, we analyze the key features, which contribute to the complexity and performance of an H.264/AVC encoder and should be considered in video transcoding to improve the performances in terms of both processing speed and visual quality.

4.2.1 Coding Features

The H.264/AVC standard employs a hybrid coding approach similar to many existing standards but different substantially in terms of the actual coding tools used. It enhances the performance of motion estimation by supporting a number of new coding features, such as multiple reference frames, variable block sizes, and quarter-pixel accuracy. These features make the motion estimation in H.264/AVC much more complex compared to that of other existing standards.

In addition, in contrast to previous standards where intra prediction is conducted in the transform domain, the intra prediction in H.264/AVC is formed in the spatial domain based on previously encoded and reconstructed blocks (refer to Chapter 2, Section 2.2.2 for more details). A number of the intra-prediction modes are intrinsically complex and require much computation time.

Besides motion estimation and intra prediction, coding mode decision is another main process that increases the computational complexity of a typical H.264/AVC encoder. To attain a high coding efficiency, the H.264/AVC standard software exhaustively examines all coding modes (intra, inter, or skipped) for each macroblock in a rate-distortion (RD) optimized fashion, minimizing a Lagrangian cost function in the form of

$$J = D + \lambda R \quad (4.1)$$

where D denotes some distortion measure between the original and the coded macroblock partitions predicted from the reference frames, R represents the number of bits required to code the macroblock difference, and λ is the Lagrange multiplier imposing a suitable rate constraint. To obtain the best coding mode, the encoder in fact performs a real coding process, including prediction and compensation, transformation, quantization, and entropy coding for all inter and intra modes, resulting in a heavy computational load.

4.2.2 H.264/AVC Rate Control

The advanced features in H.264/AVC make it difficult and inefficient to employ the existing rate control schemes of other standards. The rate control adopted

in the H.264/AVC reference software uses an adaptive frame-layer rate control scheme based on a linear prediction model [147].

In the frame-layer rate control, the target buffer bits T_{buf} allocated for the j -th frame is determined according to the target buffer level $Tbl(n_j)$, the actual buffer occupancy $B_c(n_j)$, the available channel bandwidth $u(n_j)$, and the frame rate F_r as follows

$$T_{buf} = \frac{u(n_j)}{F_r} + \gamma(Tbl(n_j) - B_c(n_j)) \quad (4.2)$$

where γ is a constant and its typical value is 0.75 when there is no B frame and 0.25 otherwise. The remaining bits are equally allocated to all not-yet-coded frames and the number of bits allocated for each frame is given by

$$T_r = \frac{R_r}{N_r} \quad (4.3)$$

where R_r is the number of remaining bits and N_r is the total number of not-yet-coded frames. Then, the target bit is a weighted combination of T_r and T_{buf}

$$T = \beta \times T_r + (1 - \beta) \times T_{buf} \quad (4.4)$$

where β is a weighting factor.

A quadratic RD model is used to calculate the corresponding quantization parameter (QP), which is then used for the RD optimization for each macroblock in the current frame. Note that the RD model requires the mean-of-absolute difference (MAD) of the residue error to estimate the QP, which is only available after the RD optimized process, thus resulting in a chicken-and-egg problem.

To solve this dilemma, the MAD required for the RD model of the current basic unit in the current frame is predicted by a linear model using the actual

MAD of the co-located position of the previous frames. Specifically, let denote the predicted MAD of the current basic unit in the current frame and the actual MAD of basic unit in the co-located position of the previous frame by MAD_{cb} and MAD_{pb} , respectively. The linear prediction model is then given by

$$MAD_{cb} = a_1 \times MAD_{pb} + a_2 \quad (4.5)$$

where a_1 and a_2 are two coefficients of the prediction model. The initial value of a_1 and a_2 are set to 1 and 0, respectively. They are updated after coding each basic unit. However, the linear model assumes the frame complexity varies gradually. If a scene change occurs, the prediction based on the information collected from the previous frames may not be accurate, which in turn may fail to obtain a suitable QP. Consequently, the number of coding bits for the current frame may not meet the target allocation bits, resulting in quality degradation.

In addition, it should be noted that the first I and P frames in the current group-of-pictures (GOP) are coded by using the QP given at the GOP layer, in which the starting QP of the first GOP is predefined and the starting QPs of other GOPs are computed based on the QPs of the previous GOP. Thus, an inappropriately predefined starting QP can affect the actual achievable bit rate and video quality. Too small a starting QP would allocate more bits to the first few frames; hence there would not be enough bits for coding other frames to closely meet the target bit rate and result in inconsistent video quality. On the other hand, too large a starting QP would result in a low quality for the first reference frame, which in turn affects the quality of the subsequent frames.

In summary, the advanced coding features in H.264/AVC can provide a better coding efficiency at the cost of increasing complexity. As many potential applications of video transcoding require the video to be transcoded in real

time or as fast as possible (e.g., video streaming over heterogeneous networks), it is therefore necessary to minimize the complexity of video transcoding without sacrificing much its coding efficiency. In this chapter, we focus on the most computationally intensive parts of H.264/AVC coding, including intra mode prediction, motion estimation, and coding mode decision, to speed up the transcoding process. Furthermore, by using the information available in the precoded video, we further enhance the H.264/AVC rate control to achieve a better quality for the transcoded video.

4.2.3 Efficient Options and Modes for Transcoding

Before discussing in details of the proposed transcoding methods, it should be noted that a large number and combination of motion-compensated prediction (MCP) modes and prediction reference frames for each macroblock are possible. Searching over all possible combinations of modes and reference frame options to maximize the overall RD performance is computationally intensive. Moreover, performance analysis conducted by Joch *et al.* [148] on fourteen common test sequences has shown that more than 80% bit savings gained by exploiting all possible macroblock partitions can be obtained using partitions not smaller than 8×8 . Furthermore, when multiple frame prediction is employed, the average bit savings for twelve test sequences are less than 5% and around 20% for the remaining two.

To examine whether the coding performance remains the same for video transcoding using H.264/AVC, we transcoded eight precoded CIF resolution H.263 sequences at 1.2 Mbits/s and 30 frames/s without using B frames (as shown in Table 4.1) to QCIF resolution H.264/AVC sequences at reduced bit

Table 4.1: PSNR results (in dB) obtained by the cascaded H.264/AVC recoding approach using four schemes with different combinations of MCP modes and reference frames.

Sequence	Target bitrate (Kbps)	Scheme I	Scheme II	Scheme III	Scheme IV
Foreman	160	32.29	33.02	33.17	33.26
Stefan	250	27.32	27.66	27.95	28.12
News	110	34.14	34.55	34.82	34.97
Tennis	230	31.44	31.85	31.96	32.01
Flower	290	33.47	33.52	33.61	33.65
Silent	110	29.78	30.73	30.86	30.95
M&D	100	34.88	35.59	35.67	35.72
Mobile	380	29.36	29.59	29.66	29.76
Average		31.58	32.06	32.21	32.31

rates using the cascaded recoding approach (i.e., the precoded videos were fully decoded and then re-encoded using the H.264/AVC standard software). Four schemes using different combinations of MCP modes and reference frames were considered: I) one mode (mode 1) and one reference frame, II) four modes (modes 1-4) and one reference frame, III) all seven modes and one reference frame, and IV) all seven modes and five reference frames.

The results show that compared with Scheme I, Scheme II can obtain an average 0.5-dB PSNR improvement. However, the performance gain by using Scheme IV compared with that of using Scheme II is only 0.25 dB on average. In addition, by exploiting all partitions smaller than 8×8 with one reference frame, Scheme III can obtain only an average 0.15-dB gain in PSNR compared with Scheme II. In our view, the much higher computation and memory cost required by exploiting all the possible coding modes and reference frame options cannot justify the small incremental performance gain for video transcoding in real-time applications. Hence, we shall limit our proposed H.264/AVC transcoding

Figure 4.1: Block diagram of the proposed video transcoder.

methods to mainly using four MCP modes (modes 1-4) and one reference frame to minimize the transcoding time.

4.3 Proposed H.263 to H.264/AVC Transcoding Methods

Fig. 4.1 shows the architecture of the proposed video transcoder. It consists of a typical H.263 decoder followed by a H.264/AVC video encoder. The precoded H.263 video is first decoded by the H.263 decoder and then re-encoded by the

H.264/AVC video encoder. For downsizing transcoding, the decoded video will be down-sampled before it is transcoded to a H.264/AVC video. In what follows, we present the three key components of our proposed H.264/AVC video transcoding methods: 1) fast intra-prediction mode selection, 2) motion vector re-estimation and inter mode selection, and 3) enhanced rate control.

4.3.1 Fast Intra-Prediction Mode Selection

1) 4×4 Luma Prediction: In intra prediction, the H.264/AVC encoder selects the mode that minimizes the sum-of-absolution difference (SAD) of 4×4 integer-transform coefficients of the difference between the prediction and the block to be coded. Although full search can obtain the optimal prediction mode, it is computationally expensive. Pan *et al.* [149] proposed a fast intra-prediction mode selection scheme based on edge direction histogram; however the computation of edge direction introduced additional complexity. Inspired by a key observation that the best prediction mode of a block is most likely in the direction of the dominant edge within that block, we propose a fast intra-prediction mode selection scheme based on the coarse edge information obtained from the integer-transform coefficients.

Note that in the DC prediction mode, the residue is computed by offsetting all pixel values of the block to be coded by the same prediction value. Thus, the AC coefficients of the 4×4 integer transform of the residue in the DC prediction mode are the same as the transform coefficients of the block to be coded. Similar to discrete cosine transform (DCT) [150], these integer-transform coefficients can be used to extract some low-level feature information.

Fig. 4.2 shows pictorially the representations for some AC coefficients of the

Figure 4.2: Pictorial representation of some 4×4 integer-transform coefficients of the difference between the prediction and the block to be coded in the DC prediction mode.

4×4 integer transform. It can be seen that the value of AC coefficient F_{01} essentially depends upon intensity difference in the horizontal direction between the left-half and the right-half of the block, gauging the strength of vertical edges. Hence, some coarse edge information, such as vertical and horizontal dominant edges, or edge orientation, can be extracted using these AC measurements in a way similar to that shown in [150] for DCT coefficients. Extending the results obtained in [150], we propose in this chapter to estimate the dominant edge orientation by

$$\theta = \tan^{-1} \left(\frac{\sum_{j=1}^3 F_{0j}}{\sum_{i=1}^3 F_{i0}} \right) \quad (4.6)$$

where θ is the angle of the dominant edge with respect to the horizontal axis, F_{i0} 's and F_{0j} 's are the integer-transform coefficients of a 4×4 block.

Given the angle θ of the dominant edge, we propose to select additional two out of nine intra-prediction modes, which have the closest orientations to

Figure 4.3: Directions of nine possible intra-prediction modes for a 4×4 block.

the edge angle θ , for a 4×4 luma prediction. Note that the edge directions of the nine possible prediction modes are shown in Fig. 4.3. Hence, if the angle θ of the dominant edge is between -26.6° and 0° , modes 1 and 6 will be selected. Therefore, together with the DC mode, we only need to perform the prediction for three modes instead of nine for a 4×4 block. As the DC mode is always included in 4×4 luma prediction, we can compute the dominant edge orientation defined by (4.6) using the AC coefficients of 4×4 integer transform of the residue in the DC prediction mode, which are available during the computation of its cost function in intra prediction [151], without incurring much additional computation. Hence, the computational complexity for 4×4 luma prediction can be reduced by a factor of 3 compared with the full search of the best intra-prediction mode.

2) 16×16 Luma Prediction: Similarly, we can obtain the edge orientations of four 8×8 blocks in a macroblock from the DCT coefficients available in the precoded video. Employing the results obtained in [150], the dominant edge

orientation of each 8×8 block can be estimated by

$$\theta = \tan^{-1} \left(\frac{\sum_{j=1}^7 G_{0j}}{\sum_{i=1}^7 G_{i0}} \right) \quad (4.7)$$

where G_{i0} 's and G_{0j} 's are the DCT coefficients of a 8×8 block obtained from the precoded video. Note that the 8×8 DCT coefficients available in the precoded video are only used to estimate the dominant edge direction for the 16×16 luma prediction. Taking the average of these edge orientations gives us the dominant edge orientation in the macroblock. Hence, in addition to the DC prediction mode which is common in homogeneous scenes, we propose to select another one out of three other possible modes based on the dominant edge orientation for a 16×16 macroblock. In this way, we can reduce the complexity of 16×16 luma prediction by a factor of 2.

Note that the fast intra prediction of the proposed transcoder is still conducted in spatial domain. It only makes use of the 4×4 integer-transform coefficients and 8×8 DCT coefficients available during the transcoding process for estimating the dominant edge direction to reduce the complexity of intra mode prediction.

4.3.2 Motion Vector Re-estimation and Inter-Mode Selection

To reduce the complexity of video transcoding, many existing methods propose to estimate the new motion vectors (MVs) required for the transcoded video directly from the MVs existing in the precoded video. In this chapter, we use a vector median filter, which has been shown able to achieve generally

the best performance [86], to resample the MVs in the precoded video. The operation of the vector median filter over a set of K corresponding MVs $V = \{mv_1, mv_2, \dots, mv_K\}$ is given by

$$\begin{aligned} mv_{\text{VM}} &= \arg \min_{mv_j \in V} \sum_{i=1}^K \|mv_j - mv_i\|_{\gamma} \\ mv' &= \mathcal{S} \times mv_{\text{VM}} \end{aligned} \quad (4.8)$$

where mv_{VM} denotes the vector median, $\|\cdot\|_{\gamma}$ is the γ -norm for measuring the distance between two MVs, mv' is the new MV required, and $\mathcal{S}$ is a 2×2 diagonal matrix downscaling the vector median mv_{VM} to suit the reduced frame size in the 2:1 downsizing transcoding. Note that in this chapter the Euclidean norm ($\gamma = 2$) is adopted for measuring the distance between two MVs.

During the encoding process, the H.264/AVC encoder needs to examine all modes and find the MV of each partition. However, a small number of available MVs for each macroblock in the H.263 precoded video make it hard to estimate the required MVs accurately. Note that in the H.264/AVC standard, the predicted MV from the neighboring macroblocks is used as the MV of the skipped mode. Thus, to enhance the transcoding performance, this predicted MV is also taken into account for estimating the new MVs.

Before we describe our proposed method, let us examine the distribution of the optimal MVs obtained by performing exhaustive search around the precoded and predicted MVs in transcoding eight well-known test sequences (listed in Table 4.1) consisting of different spatial details and motion contents. Table 4.2 shows the average and cumulative percentages of the optimal MV distribution around either the precoded or the predicted MV (i.e., the one that achieves a smaller SAD is selected as the new search center). For visualization, Fig. 4.4 also

Table 4.2: Average and cumulative percentages of the optimal MV distribution measured at different absolute distances from the new search center in eight test sequences.

Total percentage at different absolute vertical/horizontal distances								
Vertical/ Horizontal	0	1	2	3	4	5	6	7
0	64.8920	7.7059	0.6248	0.4464	0.2391	0.1733	0.1875	0.2691
1	9.3550	5.1418	0.5633	0.2701	0.1336	0.0715	0.0735	0.0884
2	0.7161	0.9548	0.3081	0.1211	0.0586	0.0376	0.0305	0.0267
3	0.4097	0.4086	0.1631	0.1704	0.0685	0.0327	0.0295	0.0277
4	0.2227	0.1856	0.0923	0.0932	0.0828	0.0404	0.0265	0.0236
5	0.1289	0.0908	0.0735	0.0361	0.0564	0.0508	0.0319	0.0227
6	0.1399	0.0852	0.0403	0.0337	0.0235	0.0421	0.0388	0.0269
7	0.1966	0.0821	0.0394	0.0420	0.0217	0.0201	0.0427	0.0459
Average and cumulative percentages at different absolute distances								
Average percentage	64.8920	22.203	3.1671	1.9893	1.1764	0.7919	0.7829	0.9755
Cumulative percentage	64.8920	87.095	90.262	92.251	93.427	94.219	95.002	95.978

shows the distribution of the optimal MVs around the new search center. The results show that most MVs obtained by exhaustive search are centered around the new search center. Specifically, around 87% of the optimal MVs are enclosed in a 3×3 window area centered around either the precoded or the predicted MV. Based on this empirical study, we propose a scheme for re-estimating the new MVs required as follows:

1) Syntax Transcoding: The MV required for each partition of each mode is simply selected from the MV in the precoded video and the predicted MV; the one that achieves a smaller SAD is selected as the new MV.

2) Downsizing Transcoding: The median MV (mv_{VM}) is first obtained from the precoded MVs for each partition of different modes as follows:

Figure 4.4: Distribution of the MVs obtained by exhaustive search around the precoded MV or the predicted MV from the neighboring macroblocks.

- Mode 1: The mv_{VM} is the downscaled median MV obtained from the four corresponding MVs in the precoded video (see (4.8)).
- Mode 2: The mv_{VM} of the upper partition is estimated from the down-scaled MVs of the two upper corresponding macroblocks; the one that achieves a smaller SAD is selected as the new MV for the upper partition. Similarly, the mv_{VM} for the lower partition is estimated from the downscaled MVs of the two lower corresponding macroblocks.
- Mode 3: Similar to Mode 2, the mv_{VM} 's of the left and right partitions are estimated from the downscaled MVs of the two left and right corresponding macroblocks, respectively.
- Mode 8×8 : The mv_{VM} for each sub-partition in an 8×8 block is simply estimated as the downscaled MV from the corresponding macroblock in

the precoded video.

The new MV required for each partition of each mode is then estimated from the mv_{VM} and the MV predicted from the neighboring blocks; the one that achieves a smaller SAD will be selected. Note that if a macroblock is intra coded in the precoded video, the zero MV will be used to re-estimate the MVs required.

Since the MVs obtained by exhaustive search are mostly centered within a small window around the re-estimated MVs obtained using the above steps, we also propose to refine the re-estimated MVs by searching a small diamond pattern centered at the re-estimated MVs [35]. To further improve the performance, the refined MVs in integer resolution can be further refined using the default quarter-pixel accuracy in H.264/AVC. To reduce the complexity, we propose to first choose the optimal inter mode based on the smallest SAD value obtained by the refined MVs in integer resolution for each mode. Thus, the MVs of only one mode need to perform the quarter-pixel refinement. Furthermore, no RD optimized process is required to choose the best inter mode, which can reduce the computational load significantly.

By using MV re-estimation, we can reduce the computational complexity for video transcoding. However, during the RD optimized process, the transcoder still needs to make a decision between intra and inter modes for each macroblock. It should be noted that the mode decision process for intra mode is computationally intensive and may cost five times of that for inter mode [152]. Based on our empirical study, we propose to adopt the MV re-estimation without using intra mode prediction for coding macroblocks in P-frames. The reason is that we can reduce the complexity notably without introducing much degradation

given that the only information available to the transcoder is the compressed video which is already lossy compressed.

4.3.3 Enhanced Rate Control for H.264/AVC Transcoding

4.3.3.1 Rate-Quantization Ratio Model

Both the H.263 and H.264/AVC reference models approximate the relation between the rate and distortion through a quadratic model, in which the number of coding bits is a quadratic function of the quantization step size. Thus, there may be a computable relation between the total number of coding bits in the precoded and transcoded videos.

To confirm, we obtained the H.263 precoded videos by encoding the Foreman and Silent sequences using an H.263 encoder with a constant QP value of 5. These precoded videos were then transcoded to H.264/AVC using various fixed QP values. Figs. 4.5 and 4.6 show the relation between the total number of coding bits per frame in the precoded and transcoded videos at various fixed QP values used in the H.264/AVC transcoder as shown in the figures. The figures show that it is likely to have a linear relation between the number of coding bits for each frame in the precoded and transcoded videos. Note that each curve in these figures contains two linear segments, in which the top-right segment representing more number of coding bits corresponds to I frames; while the bottom-left segment denoting less number of coding bits corresponds to P frames. It can be seen that the slopes of the two segments are not the same and vary for different QPs, thus suggesting the linear relation could be different for

Figure 4.5: Relation between the number of coding bits in the precoded and transcoded videos by transcoding with various fixed QPs used in the H.264/AVC transcoder for the Foreman sequence.

Figure 4.6: Relation between the number of coding bits in the precoded and transcoded videos by transcoding with various fixed QPs used in the H.264/AVC transcoder for the Silent sequence.

I and P frames and depends on the quantization step sizes of the precoded and transcoded videos.

To justify the above argument, we transcoded a number of precoded H.263 test sequences to H.264/AVC using different constant QPs. Fig. 4.7 shows the relation between the average ratio of total number of coding bits and the quantization step size ratio between the precoded and transcoded videos for I and P frames for five test sequences. The results show that the ratio of total number of coding bits between the precoded and transcoded videos likely depends on the quantization step size ratio and could be nearly constant for different video contents.

In this chapter, we propose to use a quadratic model to approximate these relations, which basically follows the trend of the actual curves. Mathematically, the proposed rate-quantization ratio (R_r - Q_r) model is given by

$$\frac{R_t^I}{R_p^I} = \frac{X_1^I}{\left(\frac{Q_t}{Q_p}\right)^2} + \frac{X_2^I}{\left(\frac{Q_t}{Q_p}\right)} + X_3^I \quad (4.9)$$

$$\frac{R_t^P}{R_p^P} = \frac{X_1^P}{\left(\frac{Q_t}{Q_p}\right)^2} + \frac{X_2^P}{\left(\frac{Q_t}{Q_p}\right)} + X_3^P \quad (4.10)$$

where $R_p^{I,P}$ and $R_t^{I,P}$ are the total numbers of coding bits, $Q_p^{I,P}$ and $Q_t^{I,P}$ are the quantization step sizes for I and P frames in the precoded and transcoded videos, respectively, and $X_1^{I,P}$, $X_2^{I,P}$, and $X_3^{I,P}$ are the model parameters. The model parameters are empirically obtained by simulation with a large number of video sequences, in which the linear least square method is used to fit the actual curves. Note that the parameters of the model are adaptively updated by using actual data points obtained during the transcoding process to make a

(a) I frame.

(b) P frame.

Figure 4.7: Relation between the average ratio of total number of coding bits and the ratio of quantization step sizes in precoded and transcoded videos.

better fit for the current video sequence.

4.3.3.2 Proposed Rate Control Method

1) Selection of Starting QP: In what follows, we propose to determine the good enough starting QP of the sequence or current GOP in order to meet closely the target bit rate. As the quality fluctuation has a negative effect on the subjective video quality, it is desirable to produce a constant quality for the transcoded video. Many experiments have indicated that using constant QP for the entire video sequence typically results in good performance, in terms of both average PSNR and consistent quality [153]. Hence, we shall choose the value of the constant QP, which can obtain the transcoded bit rate as close to the target bit rate as possible, as starting QP.

Let Q_t be the quantization step size for transcoding the remaining video in order to have the number of transcoded bits close to the number of remaining bits R_t . By using the proposed model, we can express the total number of transcoded bits with the use of a constant Q_t as

$$R_t = \sum_{k=j}^N \left(\frac{X_1}{\left(\frac{Q_t}{Q_p^k}\right)^2} + \frac{X_2}{\left(\frac{Q_t}{Q_p^k}\right)} + X_3 \right) \times R_p^k \quad (4.11)$$

where Q_p^k and R_p^k are the quantization step size and the total number of coding bits of the k -th frame in the precoded video, j is the frame number of the first frame in the current GOP, N is the total number of frames, and X_1 , X_2 , and X_3 are the corresponding model parameters depending on the picture type (I or P frame) of the k -th frame. Hence, Q_t can be obtained by solving the above quadratic equation. The starting QP of the sequence or current GOP is

determined as the nearest integer in the quantization table that corresponds to the quantization step size Q_t .

2) Allocation of Frame Bits: As mentioned earlier, H.264/AVC rate control computes the target number of bits per frame by allocating the number of remaining bits to all not-yet-coded frames equally. However, in order to achieve consistently good video quality over the entire sequence, a bit allocation scheme should take into consideration the frame complexity. The basic idea is to allocate fewer bits to less complex frames in order to save more bits for more complex frames. In this chapter, we use the number of coding bits and quantization step size in the precoded video to measure the complexity S_k of the k -th frame as

$$S_k = R_p^k \times Q_p^k \quad (4.12)$$

Hence, instead of allocating bits equally as (4.3), we propose to allocate the number of remaining bits to all not-yet-coded frames proportionally according to the frame complexity. Thus, the number of bits allocated for the k -th frame T_r^k can be computed as

$$T_r^k = R_r \times \frac{S_k}{\sum_{i=k}^N S_i} \quad (4.13)$$

The final target bit rate is then computed using (4.4).

3) Determination of Frame QP: After target bit allocation, it is important to determine the corresponding QP to meet exactly the target bit budget. However, the RD model in the existing rate control scheme may fail to determine the correct QP due to inaccurate prediction of MAD in the event of abrupt change in frame complexity. In this chapter, we propose to use the R_r - Q_r model to

determine the QP at frame level.

Similar to (4.11), the quantization step size Q_t^k for the k -th frame can be easily determined by solving

$$T_t^k = \left(\frac{X_1}{\left(\frac{Q_t^k}{Q_p^k}\right)^2} + \frac{X_2}{\left(\frac{Q_t^k}{Q_p^k}\right)} + X_3 \right) \times R_p^k \quad (4.14)$$

where T_t^k is the target number of bits for the k -th frame obtained from (4.4).

4.4 Experimental Results

To evaluate the performance of the proposed transcoding methods, our test sequences include eight popular CIF resolution (352×288) sequences, as shown in Table 4.3, which were precoded by using the Test Model 8 (TMN8) H.263 encoder [154]. In our simulation, the proposed transcoding methods were implemented on the reference software H.264/AVC JM 7.4 [155]. For each test sequence, we set the frame rate to 30 frames/s and selected the bit rate of 1.8 Mbps for the precoded videos so that there was no skipped frame in the precoded and transcoded videos. For performance comparison, we kept the bit rate constant when transcoding each sequence using different methods. The GOPs of each precoded and transcoded sequences consisted of one I frame followed by fourteen P frames. During downsizing transcoding, each precoded frame was reconstructed and downsized in spatial domain using bi-cubic interpolation. To suppress aliasing artifacts, a typical Gaussian-type low-pass filter was also applied prior to the downsizing operation. For objective comparison, the PSNR of each transcoded video was computed with respect to the original uncompressed

video with downscaling (for downsizing transcoding) or without downscaling (for syntax transcoding) to the same frame size.

In the first set of experiments, eight test sequences precoded in H.263 format were transcoded to H.264/AVC using only the MV re-estimation method proposed in Section 4.3.2 with four MCP modes (modes 1-4) and one reference frame to compare with the cascaded recoding (RC) approach using seven MCP modes and one reference frame. For comparison, two schemes of different mode options for coding macroblocks in P frames were considered: I) using both intra and inter modes, and II) using only inter mode. Table 4.3 shows the PSNR and complexity results in terms of total encoding time based on our implementation. The results show that incurring much lower computational costs, both schemes can perform comparably to the H.264/AVC RC scheme. Specifically, the average PSNR results obtained by the proposed schemes are only about 0.35-dB inferior to that obtained by the H.264/AVC RC scheme both with and without downscaling, while the total encoding time of Scheme II is reduced by a factor of about 6 compared with that of the H.264/AVC RC scheme. It should be noted that by using quarter-pixel refinement, we can achieve about 0.9-dB and 1.3-dB improvements in PSNR both with and without downscaling respectively in both schemes. In addition, without using intra mode in P frames, Scheme II can reduce the computational cost substantially while the performance is only a little worse than that of Scheme I. Furthermore, compared with Scheme I (without quarter-pixel refinement), Scheme II (with quarter-pixel refinement) not only is much less computationally expensive, it can also obtain an average of more than 1-dB performance gain in PSNR.

In another set of experiments, we repeated the first experiment by using

Table 4.3: PSNR results and the total encoding times obtained by transcoding H.263 sequences using the cascaded H.264/AVC recoding (RC) method, the MV re-estimation method proposed in Section 4.3.2, with or without quarter-pixel refinement (refn.).

Sequence	Target bitrate (Kbps)	Transcoded frame size	Scheme I		Scheme II		H.264/AVC RC refn.
			No refn.	refn.	No refn.	refn.	
Foreman	800	352×288	34.94	36.43	34.42	36.12	36.67
	155	176×144	31.57	32.74	31.35	32.61	33.08
M&D	600	352×288	39.83	40.60	39.67	40.51	40.66
	130	176×144	37.17	38.05	37.13	38.01	38.12
News	600	352×288	38.64	39.33	38.33	39.17	39.50
	130	176×144	34.27	35.10	34.20	35.04	35.32
Silent	800	352×288	35.08	35.36	34.90	35.26	35.43
	155	176×144	33.67	34.09	33.55	34.02	34.22
Stefan	1000	352×288	28.28	31.55	27.48	31.05	31.86
	300	176×144	26.80	28.83	26.46	28.66	29.36
Tennis	900	352×288	31.12	31.47	30.82	31.33	31.63
	340	176×144	35.02	35.38	34.88	35.31	35.66
Mobile	1200	352×288	25.97	27.88	25.66	27.83	28.16
	600	176×144	34.33	35.41	34.31	35.41	35.60
Flower	800	352×288	29.05	29.81	29.00	29.80	29.90
	250	176×144	30.84	31.44	30.84	31.45	31.48

PSNR (in dB)

Sequence	Transcoded frame size	Scheme I		Scheme II		H.264/AVC RC refn.
		No refn.	refn.	No refn.	refn.	
Foreman	352×288	955	1,144	220	341	1,924
	176×144	217	270	54	82	462
M&D	352×288	949	1,092	210	319	1,805
	176×144	228	277	57	77	449
News	352×288	997	1,130	215	326	1,866
	176×144	231	285	55	79	452
Silent	352×288	1,082	1,214	227	339	1,944
	176×144	243	288	60	85	483
Stefan	352×288	948	1,134	224	343	1,904
	176×144	240	285	62	86	488
Tennis	352×288	1,341	1,416	271	432	2,577
	176×144	269	298	64	85	496
Mobile	352×288	1,416	1,588	353	488	2,551
	176×144	400	451	97	117	642
Flower	352×288	1,158	1,291	239	342	2,402
	176×144	278	306	60	82	475

Total encoding time (in second)

both transcoding methods proposed in Sections 4.3.1 and 4.3.2. Similar to the first experiment, both schemes with and without using intra mode in P frames were considered, which are denoted as Scheme I' and Scheme II', respectively. Note that the only difference between these schemes and those in Table 4.3 is that the proposed intra-prediction mode selection scheme (IPMS) was used to find the best mode in intra prediction instead of the full search IPMS. Table 4.4 shows the average PSNR results and total encoding times. Observably, using the proposed IPMS with fewer intra-prediction modes for transcoding (Scheme I' and Scheme I'') can reduce the encoding complexity significantly compared with the full search IPMS (Scheme I and Scheme II) while introducing only about 0.1-dB loss in PSNR.

Figs. 4.8 and 4.9 show the frame-to-frame PSNR results of the News and Foreman sequences obtained by using Scheme I' and Scheme II' in comparison with the H.264/AVC RC scheme. It should be noted that the proposed method can perform modestly inferior compared to the cascaded H.264/AVC RC scheme and the differences are uniformly distributed over the entire sequences. For visual comparison, Figs. 4.10 and 4.11 show sample frames of the News and Foreman sequences obtained by the proposed method. The figures show that the proposed method can achieve a good perceived video quality, in terms of sharpness and blocking artifacts, compared with the cascaded recoding scheme, and perform visually about the same as the cascaded recoding scheme.

To evaluate the performance of the proposed rate control method when used together with the proposed fast transcoding methods, the Foreman sequence was transcoded to H.264/AVC using the proposed R_r - Q_r model to determine the starting QP at different target bit rates that were generated by transcoding

Table 4.4: PSNR results and the total encoding times obtained by transcoding H.263 sequences using the cascaded H.264/AVC recoding (RC) method, and the proposed H.264/AVC transcoding methods in Sections 4.3.1 and 4.3.2, with or without quarter-pixel refinement (refn.).

Sequence	Target bitrate (Kbps)	Transcoded frame size	Scheme I'		Scheme II'		H.264/AVC RC refn.
			No refn.	refn.	No refn.	refn.	
Foreman	800	352×288	34.82	36.35	34.39	36.08	36.67
	155	176×144	31.46	32.65	31.25	32.52	33.08
M&D	600	352×288	39.78	40.57	39.63	40.49	40.66
	130	176×144	37.10	37.97	37.06	37.92	38.12
News	600	352×288	38.52	39.24	38.28	39.09	39.50
	130	176×144	34.10	34.90	34.06	34.88	35.32
Silent	800	352×288	35.02	35.32	34.87	35.24	35.43
	155	176×144	33.55	33.96	33.46	33.91	34.22
Stefan	1000	352×288	28.10	31.47	27.42	30.98	31.86
	300	176×144	26.68	28.76	26.36	28.60	29.36
Tennis	900	352×288	31.06	31.43	30.79	31.30	31.63
	340	176×144	34.97	35.35	34.83	35.26	35.66
Mobile	1200	352×288	25.87	27.83	25.63	27.80	28.16
	600	176×144	34.29	35.38	34.28	35.38	35.60
Flower	800	352×288	28.99	29.75	28.96	29.75	29.90
	250	176×144	30.72	31.30	30.71	31.32	31.48

PSNR (in dB)

Sequence	Transcoded frame size	Scheme I'		Scheme II'		H.264/AVC RC refn.
		No refn.	refn.	No refn.	refn.	
Foreman	352×288	597	752	193	314	1,924
	176×144	138	162	44	72	462
M&D	352×288	584	707	185	292	1,805
	176×144	135	166	43	73	449
News	352×288	611	749	185	295	1,866
	176×144	135	166	42	67	452
Silent	352×288	656	773	192	306	1,944
	176×144	146	178	43	75	483
Stefan	352×288	586	745	200	303	1,904
	176×144	146	178	49	77	488
Tennis	352×288	641	767	194	303	2,577
	176×144	160	188	49	77	496
Mobile	352×288	726	874	236	359	2,551
	176×144	278	259	74	102	642
Flower	352×288	692	788	199	302	2,402
	176×144	157	198	48	72	475

Total encoding time (in second)

Figure 4.8: Frame-to-frame PSNR results of the News sequence obtained by the proposed H.264/AVC transcoding methods and the H.264/AVC recoding (RC) method.

Figure 4.9: Frame-to-frame PSNR results of the Foreman sequence obtained by the proposed H.264/AVC transcoding methods and the H.264/AVC recoding (RC) method.

Original precoded frame (37.12 dB)

Scheme I' (35.71 dB)

Scheme II' (35.35 dB)

H.264/AVC RC (35.97 dB)

Figure 4.10: Sample frames from the News sequence transcoded from the precoded H.263 video by the proposed syntax transcoding methods with quarter-pixel refinement and the cascaded H.264/AVC recoding (RC) method.

Original precoded frame (39.47 dB)

Scheme I' (37.78 dB)

Scheme II' (37.43 dB)

H.264/AVC RC (38.12 dB)

Figure 4.11: Sample frames from the Foreman sequence transcoded from the precoded H.263 video by the proposed syntax transcoding methods with quarter-pixel refinement and the cascaded H.264/AVC recoding (RC) method.

Figure 4.12: PSNR results (in dB) and the differences between the target and achieved bit rates obtained by transcoding the Foreman sequence using various starting QPs at a given target bit rate.

with different constant QPs. The simulation shows that the proposed model is able to select the starting QP close to the constant QP, which was used to generate each target bit rate. Fig. 4.12 shows the PSNR results and the difference between the target and achieved bit rates for various starting QPs. Obviously, the starting QPs in the bottom-right area are preferred, where the differences between the target and achieved bit rates are close to zero and the PSNRs are high enough. The results show that by using the proposed method, the transcoder can determine a reasonably good starting QP (red marker) in order to meet closely the given target bit rate and achieve good video quality. Furthermore, the proposed method can provide more consistent video quality in terms of low PSNR fluctuation (measured by the standard deviation (σ) of the PSNR results over each entire sequence) as shown in Table 4.5. Note that Table 4.5 tabulates the numerical results of using several QP values around the

Table 4.5: PSNR results (in dB), standard deviations of the PSNR results (σ), and achieved bit rates for the Foreman sequence using different starting QPs at a given target bit rate.

Starting QP	Target bitrate (Kbps)	Achieved bitrate (Kbps)	PSNR (dB)	σ
16	84.83	86.36	32.90	3.51
20	84.83	85.55	34.44	1.56
24	84.83	84.67	34.74	0.95
28	84.83	84.70	34.72	0.88
32	84.83	84.62	34.65	0.86
36	84.83	84.18	34.38	1.21

Note: The starting QP obtained by the proposed model is 28.

selected one (bottom-right area in Fig. 4.12); these QPs generally result in too large difference between target and achieved bit rates or too low PSNR.

In the last set of experiments, we transcoded six QCIF resolution (176×144) sequences at 15 frames/s by the cascaded H.264/AVC RC approach using seven MCP modes and one reference frame and the fast transcoding methods proposed in Sections 4.3.1 and 4.3.2 with both the existing H.264/AVC and proposed rate control methods using four MCP modes (modes 1-4) and one reference frame. The results in Table 4.6 show that the standard deviation of the PSNR performance obtained using the H.264/AVC RC method is slightly better than that of using the proposed fast transcoding methods with the existing H.264/AVC rate control. However, by using the proposed rate control method for transcoding, the quality of transcoded video can be further enhanced. Specifically, compared with the H.264/AVC RC method, the transcoded video obtained by using the proposed rate control method can meet the target bit rate more accurately; furthermore, the standard deviation of the PSNR performance is lower than that obtained by the H.264/AVC RC method, which implies a more consistent visual quality over the entire sequence.

Table 4.6: PSNR results (in dB), standard deviations of the PSNR results (σ), and actual bit rates obtained by the H.264/AVC recoding (RC) method, the proposed fast transcoding methods in conjunction with the existing and the proposed H.264/AVC rate control methods.

Sequence	Target	Proposed fast transcoding methods						H.264/AVC		
	bitrate	Existing rate control			Proposed rate control			cascaded recoding		
	(Kbps)	Actual	PSNR	σ	Actual	PSNR	σ	Actual	PSNR	σ
		bitrate	(dB)		bitrate	(dB)		bitrate	(dB)	
	(Kbps)	(Kbps)			(Kbps)			(Kbps)		
Foreman	65.17	64.87	33.81	1.015	65.15	33.83	0.652	64.58	34.22	0.923
	111.05	111.80	35.52	0.792	111.17	35.57	0.543	111.72	35.95	0.712
News	31.89	32.81	32.92	1.167	31.92	32.95	0.915	32.75	33.27	1.095
	55.46	56.38	35.67	1.093	55.53	35.60	0.835	56.41	36.06	0.964
Silent	60.38	60.74	34.58	0.787	60.36	34.64	0.493	60.59	34.79	0.792
	109.59	110.60	36.51	0.597	109.64	36.56	0.326	108.62	36.78	0.543
Stefan	140.28	140.73	29.03	1.433	140.34	29.11	0.975	140.84	29.75	1.355
	292.04	292.98	32.34	1.281	292.19	32.30	0.713	292.76	33.13	1.153
Tennis	73.49	74.11	30.55	2.191	73.62	30.57	1.793	72.89	30.84	2.018
	136.60	136.92	32.81	1.712	136.58	32.87	1.433	136.89	33.14	1.582
Mobile	194.44	195.54	28.47	1.132	194.66	28.54	0.847	195.32	28.66	1.087
	388.00	388.54	31.03	0.984	388.13	31.17	0.652	388.61	31.25	0.941

Figs. 4.13 and 4.14 show the frame-to-frame PSNR results of the Foreman and Silent sequences obtained by the H.264/AVC RC method and the proposed fast transcoding methods together with the enhanced rate control. Not surprisingly, the fluctuation of PSNR obtained by transcoding with the proposed rate control method is less than that of the H.264/AVC RC method. This can be explained by the fact that our method allocates bits per frame based on the frame complexity. Furthermore, we used the proposed R_r - Q_r model to determine more accurate QP for each frame rather than using MAD prediction that can be inaccurate in the event of abrupt change in frame complexity.

Figure 4.13: PSNR performance (in dB) of the transcoded videos of the Foreman sequence obtained by the H.264/AVC rate control method and the proposed enhanced rate control.

Figure 4.14: PSNR performance (in dB) of the transcoded videos of the Silent sequence obtained by the H.264/AVC rate control method and the proposed enhanced rate control.

4.5 Summary

We have proposed in this chapter an efficient method for H.263 to H.264/AVC video transcoding. Besides using a vector median filter for motion re-estimation, we have also proposed a fast intra-prediction mode selection scheme based on the coarse edge information obtained from integer-transform coefficients. In addition, an enhanced rate control method is proposed to improve the transcoded video quality. The proposed rate control method uses a quadratic model for selecting quantization parameters at the sequence and frame levels together with a new frame-layer bit allocation scheme based on the side information from the precoded video. The experimental results show the accuracy of the model and the effectiveness of the proposed methods. In particular, the PSNR obtained by the proposed methods is only about 0.35-dB inferior to that obtained by the cascaded H.264/AVC recoding scheme, while the total transcoding time can be reduced by a factor of 6. Furthermore, the proposed rate control method can meet the target bit rate more accurately and provide more consistent video quality compared with that of existing H.264/AVC rate control scheme.

Chapter 5

Blind Video Enhancement from Multiple Compressed Copies

In this chapter, we cover the topic of quality enhancement for compressed videos using post-processing techniques. We address a new research problem by blindly enhancing the quality of the video reconstructed from multiple compressed copies of the same video content. The main objective is to reconstruct a video that achieves better quality than any of the available copies. A narrow quantization constraint set together with the Laplacian and Cauchy approximations for the distribution of each AC transform coefficient are proposed to enhance the reconstructed video.

This chapter is organized as follows. After presenting the introduction in Section 5.1, Section 5.2 briefly reviews some key features of transform-based coding in the latest H.264/AVC standard exploited in this chapter. Section 5.3 formulates the problem of blindly enhancing the video content from multiple compressed copies and describes the proposed enhancement method with the

assumption that the temporal resolutions of the available copies are well aligned. An effective method for temporal registration of multiple compressed copies is presented in Section 5.4. Mathematical analysis and experimental results to justify the performance of the proposed method are presented in Sections 5.5 and 5.6, respectively. In Section 5.7, we conclude the chapter by summarizing our main contributions.

5.1 Introduction

Achieving good energy compaction over a wide class of visual signals, block-based coding transform is commonly adopted in most popular video compression standards to exploit the spatial correlation of video signals. While block-based coding can attain good quality at high bit rates, it often suffers from undesirable coding artifacts (such as blocking artifact, ringing noise, and corner outliers) at moderate to low bit rates. These coding artifacts are mainly due to the error introduced by the quantization/dequantization process, which may result in severe loss in visual quality and fidelity of the reconstructed video.

To alleviate this problem, post-processing is one of the most promising solutions as it can improve the video quality without the need of changing the encoder structure. Many post-processing techniques have been proposed to reduce the quantization artifacts of block-based coding. These include block-boundary post-filtering techniques to smooth the discontinuous in either spatial [90,94] or transform domain [21,93] such as adaptive filtering and wavelet-based filtering. Also proposed are more sophisticated methods that enhance the reconstructed video by using image/video restoration techniques such as iterative methods

based on the theory of projection onto convex sets (POCS) or constrained minimization [112, 114, 116], maximum a posterior probability estimation approach (MAP) [108], and regularized image/video restoration [110, 113]. These methods consider the compressed images/videos to be distorted by a codec system and apply restoration techniques to reduce the quantization noises and coding artifacts.

With the development of network and communication techniques as well as the popularity of video-centric websites such as YouTube, Facebook, and Google Video, delivery of visual signals over the network has become more and more popular. Given the phenomenal rate at which image and video contents are being generated and distributed, we can now easily obtain many copies of the same video content with different levels of visual quality. For example, different people may record the same interesting soccer match or a piece of news from a television channel and encode it in different formats or using different coding parameters to meet their constraints (e.g., transmission bandwidth, storage capacity, etc.) before sharing it over the network. Similarly, one can gain access to many copies of movie trailers or video clips extracted from DVDs, which have exactly the same content but different visual quality.

Employing the existing post-processing techniques, one can possibly enhance the quality of each of these compressed copies independently from the other copies. However, as the original source video or information on the video quality is not always available, how to obtain the best video from these multiple compressed copies becomes an interesting problem. The problem shares some similarity with the well-known super resolution (SR) restoration problem, which has been addressed intensively in the literature [156–161]. For example, Gun-

turk *et al.* [158, 159] proposed to reconstruct high-resolution images by using multiple neighboring low-resolution frames of compressed videos. It should be noted that the restoration or enhancement of high-resolution images in SR requires a set of low-resolution observations, which usually contain different but related views of the scene (e.g., images taken from different cameras, view angles, illumination conditions, or even a sequence of frames from a video). What we consider here is, however, to enhance the video quality from multiple compressed copies of the same content (i.e., no spatial variations) with different levels of quantization noise.

In this chapter, we address this new research problem by blindly enhancing the quality of the video reconstructed from multiple compressed copies of the same visual content, where the existing post-processing techniques may no longer be suitable nor effective as they usually consider only a single compressed video. Our aim is to reconstruct a video that achieves better quality than any of the available copies. The proposed method is considered to be a “blind” approach as the original source video is not available, and this makes the problem particularly challenging as we don’t definitively know which of the multiple copies, which frame of a copy, and which region of a frame have the best quality.

Specifically, we propose to reconstruct each coefficient of the enhanced video in the transform domain by using a narrow quantization constraint set derived from the multiple compressed copies and in which the exact value of the coefficient should lie. In addition, a Laplacian or Cauchy distribution model is utilized to further reduce the distortion of each AC transform coefficient. Analytical and experimental results show that the video reconstructed by the proposed method generally yields a distortion smaller than that of any of the compressed copies

available. In many scenarios, the proposed method can attain a notable gain in terms of average peak-signal-to-noise ratio (PSNR) compared to the best video from the multiple compressed copies.

5.2 Brief Overview of H.264/AVC Transform and Quantization

In this section, we shall review some key features of the transform-based coding in the state-of-the-art video coding standard H.264/AVC, which will be exploited in this chapter to address the aforementioned problem.

In essence, the existing video coding standards support intra coding and inter coding. While inter coding employs temporal prediction (motion compensation) from previously encoded pictures, intra coding only uses the information contained in the picture itself. The prediction residue (either intra or inter), which is the difference between the original and the predicted pictures, is then transformed, quantized, and entropy coded. In H.264/AVC, the transformation is applied to 4×4 blocks and instead of using the DCT-based transform like previous standards (e.g., H.263 and MPEG-1/2/4), a separable integer transform with basically the same properties as a 4×4 DCT is employed to avoid the mismatch between the encoder and decoder.

Let $\mathbf{w}$ denote a 4×4 block. The 4×4 integer transform coefficients of $\mathbf{w}$ are defined in H.264/AVC as

$$\mathbf{W} = (\mathbf{C}_f \mathbf{w} \mathbf{C}_f^T) \otimes \mathbf{E}_f \quad (5.1)$$

where $\otimes$ represents point-to-point multiplication (e.g., each element of $(\mathbf{C}_f \mathbf{w} \mathbf{C}_f^T)$ is multiplied by the element in the same position of matrix $\mathbf{E}_f$), $\mathbf{C}_f$ is the forward transformation matrix, and $\mathbf{E}_f$ is the forward post-scaling factor matrix, which are defined as [162]

$$\mathbf{C}_f = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}; \quad \mathbf{E}_f = \begin{bmatrix} \frac{1}{4} & \frac{1}{\sqrt{40}} & \frac{1}{4} & \frac{1}{\sqrt{40}} \\ \frac{1}{\sqrt{40}} & \frac{1}{10} & \frac{1}{\sqrt{40}} & \frac{1}{10} \\ \frac{1}{4} & \frac{1}{\sqrt{40}} & \frac{1}{4} & \frac{1}{\sqrt{40}} \\ \frac{1}{\sqrt{40}} & \frac{1}{10} & \frac{1}{\sqrt{40}} & \frac{1}{10} \end{bmatrix} \quad (5.2)$$

Note that if the macroblock is coded in the 16×16 intra-prediction mode, the DC coefficients of the 4×4 luma residue blocks will be transformed again using a 4×4 Hadamard transform to decorrelate the DC coefficients before the quantization process (see [162] for details).

Let Z_{ij} denote the quantized coefficient value and $\widehat{W}_{ij}$ denote the dequantized coefficient value of W_{ij} . The quantization process in H.264/AVC is defined as follows

$$\begin{aligned} Z_{ij} &= \text{sgn}(W_{ij}) \times \left\lfloor \frac{|W_{ij}|}{Q_{ij}} + g \right\rfloor \\ \widehat{W}_{ij} &= Z_{ij} \times Q_{ij} \end{aligned} \quad (5.3)$$

where Q_{ij} is the quantization step size, g is the rounding control parameter, $\lfloor \cdot \rfloor$ is the floor operator that rounds to the nearest integer towards minus infinity, and $\text{sgn}(\cdot)$ returns the sign of a signal. In the implementation of the H.264/AVC reference software [155], $g = 1/3$ for intra frames and $g = 1/6$ for inter frames. It should also be noted that the post-scaling operation is incorporated together with the forward quantizer in the reference software to avoid rounding errors in

Table 5.1: Quantization step size conversion function

$\text{mod}(QP, 6)$	QP2QSTEP
0	0.6250
1	0.6875
2	0.8125
3	0.8750
4	1.0000
5	1.1250

the transformation process. The quantization step size Q_{ij} is determined by a quantization parameter (QP), which is calculated by the rate control algorithm and may be different for each macroblock. The quantization step size will double in size for every increment of 6 in QP and increase by 12.5% for each increment of 1 in QP. Mathematically, Q_{ij} can be computed from QP as

$$Q_{ij} = \text{QP2QSTEP}(\text{mod}(QP, 6))2^{\lfloor QP/6 \rfloor} \quad (5.4)$$

where the QP2QSTEP function is defined in Table 5.1, and $\text{mod}(M, N)$ is the remainder of integer division of M by N .

The inverse transform is given by [162]

$$\hat{\mathbf{w}} = \mathbf{C}_i^T (\widehat{\mathbf{W}} \otimes \mathbf{E}_i) \mathbf{C}_i \quad (5.5)$$

where $\mathbf{C}_i$ is the inverse transformation matrix, and $\mathbf{E}_i$ is the inverse post-scaling

factor matrix, which are defined as

$$\mathbf{C}_i = \begin{bmatrix} 1 & 1 & 1 & \frac{1}{2} \\ 1 & \frac{1}{2} & -1 & -1 \\ 1 & -\frac{1}{2} & -1 & 1 \\ 1 & -1 & 1 & -\frac{1}{2} \end{bmatrix}; \quad \mathbf{E}_i = \begin{bmatrix} \frac{1}{4} & \frac{1}{\sqrt{10}} & \frac{1}{4} & \frac{1}{\sqrt{10}} \\ \frac{1}{\sqrt{10}} & \frac{2}{5} & \frac{1}{\sqrt{10}} & \frac{2}{5} \\ \frac{1}{4} & \frac{1}{\sqrt{10}} & \frac{1}{4} & \frac{1}{\sqrt{10}} \\ \frac{1}{\sqrt{10}} & \frac{2}{5} & \frac{1}{\sqrt{10}} & \frac{2}{5} \end{bmatrix} \quad (5.6)$$

5.3 Problem Formulation and Proposed Method

We can formulate the problem considered as follows: given K different compressed copies of the same video content, let $Y_k(i)$ and $Q_k(i)$ denote the quantized (residual) value and the corresponding quantization step size of the i -th integer transform coefficient in a video frame of the k -th copy, respectively, and, if any, let $P_k(i)$ be the corresponding intra-prediction value for intra mode or motion-compensated prediction value for inter mode in the integer transform domain. Note that the prediction values from the compressed videos are only available in the spatial domain. Hence, $P_k(i)$ is obtained by using the integer transform as defined in (5.1). The decoded value $\hat{X}_k(i)$ of the i -th integer transform coefficient in the k -th copy is computed as

$$\hat{X}_k(i) = Y_k(i) \times Q_k(i) + P_k(i) \quad (5.7)$$

The distortion of the decoded frame of the k -th copy in terms of mean square error (MSE) can be computed in the transform domain according to Parseval's

Theorem as follows

$$mse_k = \frac{1}{N} \sum_{i=1}^N (X(i) - \hat{X}_k(i))^2 \quad (5.8)$$

where N is the total number of pixels in the video frame, and $X(i)$ denotes the value of the i -th integer transform coefficient in the original frame.

Let $\hat{X}(i)$ be the estimated value of the i -th coefficient in the enhanced reconstructed video. The objective is to find $\hat{X}(i)$ given $\hat{X}_k(i)$'s and $Q_k(i)$'s from K multiple compressed copies of the same video content such that the distortion of the reconstructed video is no larger than that of any given copy, i.e.,

$$mse = \frac{1}{N} \sum_{i=1}^N (X(i) - \hat{X}(i))^2 \leq mse_k \quad \forall 1 \leq k \leq K \quad (5.9)$$

As quantization is a many-to-one mapping operation, the quantized value and quantization step size of each coefficient will specify an interval, referred to as the quantization constraint set (QCS), in which the exact value of the coefficient should lie. Different compressed copy will possibly give a different QCS for each coefficient in the video frame. In the simple dead-zone scalar quantization defined by (5.3), it is easy to see that each QCS of the i -th integer

transform coefficient reveals that $X(i) \in S_k(i) = [\alpha_k(i), \beta_k(i)]$ where

$$\begin{cases} \alpha_k(i) = \hat{X}_k(i) - (1-g)Q_k(i) \\ \beta_k(i) = \hat{X}_k(i) + (1-g)Q_k(i) \end{cases} \quad \text{if } Y_k(i) = 0; \\ \begin{cases} \alpha_k(i) = \hat{X}_k(i) - gQ_k(i) \\ \beta_k(i) = \hat{X}_k(i) + (1-g)Q_k(i) \end{cases} \quad \text{if } Y_k(i) > 0; \quad (5.10) \\ \begin{cases} \alpha_k(i) = \hat{X}_k(i) - (1-g)Q_k(i) \\ \beta_k(i) = \hat{X}_k(i) + gQ_k(i) \end{cases} \quad \text{if } Y_k(i) < 0.$$

Given K compressed copies, we can have up to K different QCSs $S_1(i), S_2(i), \dots, S_K(i)$ in which the exact value of the i -th integer transform coefficient should lie. Thus, if the frames of these K different compressed copies are well aligned, we can obtain a narrow QCS for $X(i)$ by taking the intersection of these sets, as follows:

$$S(i) = [\alpha(i), \beta(i)] = \bigcap_{k=1}^K S_k(i) \quad (5.11)$$

where $\begin{cases} \alpha(i) = \max \{ \alpha_k(i) \mid 1 \leq k \leq K \} \\ \beta(i) = \min \{ \beta_k(i) \mid 1 \leq k \leq K \} \end{cases}$

Eq. (5.11) shows that by having multiple compressed copies, the size of the QCS for each integer transform coefficient can likely be reduced. The reduction in the QCS size allows us to estimate a more accurate integer transform coefficient and thus reconstruct a video frame with a lower distortion. Ideally, we can obtain the exact integer transform coefficient if the QCS becomes a scalar, a scenario that rarely occurs. Hence, we propose in this chapter to reconstruct each integer transform coefficient by using the corresponding narrow QCS and

justify that by doing so the constraints in (5.9) can be satisfied.

Since each integer transform coefficient is quantized independently by a quantization step size, minimizing the MSE subject to the constraints specified in (5.9) is equivalent to minimizing the distortion caused by each integer transform coefficient. In order to have a quantization that better fits to a non-uniform distribution of the integer transform coefficient over the QCS, H.264/AVC decoder uses the rounding control parameter g in (5.3) to control the position of the reconstructed value inside the QCS interval. Due to the non-symmetric distribution, the reconstructed value is not located in the center of the corresponding QCS like the previous coding standards such as H.263 and MPEG-1/2/4. A fixed value of g smaller than half of the QCS size is used to reduce the quantization error. However, to achieve the optimal quantization error, the reconstructed value should be adaptively decided based on the probability distribution of the integer transform coefficient over the corresponding QCS.

Consider the i -th integer transform coefficient $X(i)$. Let $f(x)$ denote the probability density function (pdf) of that integer transform coefficient. Given a QCS $S(i) = [\alpha(i), \beta(i)]$ of $X(i)$, the average distortion incurred by reconstructing the coefficient as $\hat{X}(i)$ can be computed as

$$\varepsilon(S(i)) = \int_{\alpha(i)}^{\beta(i)} (x - \hat{X}(i))^2 f_S(x) dx \quad (5.12)$$

where $f_S(x)$ is the pdf of $X(i)$ over the QCS $S(i)$, which has the form of $f_S(x) = cf(x)$. Here, c is a constant so that $\int_S cf(x)dx = 1$. It follows that

$c = \frac{1}{\int_{\alpha(i)}^{\beta(i)} f(x)dx}$ and substitute into (5.12), we have

$$\varepsilon(S(i)) = \frac{\int_{\alpha(i)}^{\beta(i)} (x - \hat{X}(i))^2 f(x)dx}{\int_{\alpha(i)}^{\beta(i)} f(x)dx} \quad (5.13)$$

To minimize (5.13), it is easy to show that (see [163] for example), the reconstructed value $\hat{X}(i)$ should be chosen as the centroid of the QCS $S(i)$, given by

$$\hat{X}(i) = \frac{\int_{\alpha(i)}^{\beta(i)} x f(x)dx}{\int_{\alpha(i)}^{\beta(i)} f(x)dx} \quad (5.14)$$

It has been shown in the previous studies that in the DCT transform domain of a natural image, while the DC coefficient can be approximated as the uniform distribution, the AC coefficient distribution can be modeled by a generalized Gaussian [164, 165] or Laplacian [166, 167] probability density function. Although the generalized Gaussian model gives the most accurate representation of the AC coefficient distribution, the Laplacian model is commonly employed due to it being more tractable both mathematically and computationally. Recently, Kamaci *et al.* [168] proposed to use the Cauchy model, which is shown as a better choice than the Laplacian model for estimating the actual probability distribution of AC coefficients. In this chapter, both Laplacian and Cauchy models will be examined for the estimation of the reconstructed video. In what follows, we present a method to estimate the parameters of both distribution models by using the decoded values from the compressed videos.

Laplacian Model Parameter Estimation: The Laplacian probability density function can be described by

$$f(x) = \frac{b}{2} e^{-b|x|} \quad (5.15)$$

where b is the distribution parameter and x is the coefficient value for a given AC frequency. If the original coefficient values are known, an estimation of parameter b could be computed using the maximum-likelihood (ML) method as

$$\hat{b}_{ML} = \operatorname{argmax} \left\{ \log \prod_{t=1}^T f(X_t(i)) \right\} \quad (5.16)$$

where $X_t(i)$ is the t -th sample of the original coefficient values for a given AC frequency and T is the number of coefficients.

To estimate the Laplacian distribution of each AC coefficient from the dequantized values, we adopt the ML method proposed in [169]. Let $\hat{X}_1(i), \hat{X}_2(i), \dots, \hat{X}_T(i)$ be the dequantized values for a given AC frequency from all the 4×4 integer transform blocks of the video frames in the K compressed copies, and $Q_1(i), Q_2(i), \dots, Q_T(i)$ be the corresponding quantization step sizes. Note that the number of dequantized values T will be the product of the number of 4×4 integer transform blocks in a video frame and the number of compressed copies K . Let $S_t(i) = [\alpha_t(i), \beta_t(i)]$ be the QCS in which the original coefficient $X_t(i)$ lies and can be computed using (5.10). The ML estimate of parameter b is given by

$$\hat{b}_{ML} = \operatorname{argmax} \left\{ \log \prod_{t=1}^T P(\hat{X}_t(i)) \right\} \quad (5.17)$$

where $P(\hat{X}_t(i))$ is the probability of the reconstructed AC coefficient being $\hat{X}_t(i)$,

and it can be computed as

$$\begin{aligned}
 P(\hat{X}_t(i)) &= \int_{\alpha_t(i)}^{\beta_t(i)} \frac{b}{2} e^{-b|x|} dx \\
 &= \begin{cases} 1 - \frac{1}{2}e^{b\alpha_t(i)} - \frac{1}{2}e^{-b\beta_t(i)} & \text{if } \alpha_t(i)\beta_t(i) < 0 \\ \frac{1}{2}\text{sgn}(\alpha_t(i)) (e^{-b|\alpha_t(i)|} - e^{-b|\beta_t(i)|}) & \text{otherwise} \end{cases} \quad (5.18)
 \end{aligned}$$

Substituting (5.18) into (5.17), we have

$$\begin{aligned}
 \hat{b}_{ML} = \text{argmax} \left\{ \sum_{t=1, \alpha_t(i)\beta_t(i) < 0}^T \log \left(1 - \frac{1}{2}e^{b\alpha_t(i)} - \frac{1}{2}e^{-b\beta_t(i)} \right) + \right. \\
 \left. \sum_{t=1, \alpha_t(i)\beta_t(i) \geq 0}^T \log \left[\frac{1}{2}\text{sgn}(\alpha_t(i)) (e^{-b|\alpha_t(i)|} - e^{-b|\beta_t(i)|}) \right] \right\} \quad (5.19)
 \end{aligned}$$

Differentiating (5.19) with respect to b , we obtain

$$\begin{aligned}
 &\sum_{t=1, \alpha_t(i)\beta_t(i) < 0}^T \frac{\beta_t(i)e^{-b\beta_t(i)} - \alpha_t(i)e^{b\alpha_t(i)}}{2 - e^{b\alpha_t(i)} - e^{-b\beta_t(i)}} + \\
 &\sum_{t=1, \alpha_t(i)\beta_t(i) \geq 0}^T \frac{|\beta_t(i)|e^{-b|\beta_t(i)|} - |\alpha_t(i)|e^{-b|\alpha_t(i)|}}{e^{-b|\alpha_t(i)|} - e^{-b|\beta_t(i)|}} = 0 \quad (5.20)
 \end{aligned}$$

whose solution can be found by using an iterative root finding algorithm.

Cauchy Model Parameter Estimation: The Cauchy probability density function can be described by

$$f(x) = \frac{1}{\pi} \frac{b}{b^2 + x^2} \quad (5.21)$$

where b is the distribution parameter. Similar to the Laplacian model, the parameter b in the Cauchy model can be estimated by the method using (5.17)

where $P(\hat{X}_t(i))$ can be computed as

$$\begin{aligned} P(\hat{X}_t(i)) &= \int_{\alpha_t(i)}^{\beta_t(i)} \frac{1}{\pi} \frac{b}{b^2 + x^2} dx \\ &= \frac{1}{\pi} \left[\tan^{-1} \left(\frac{\beta_t(i)}{b} \right) - \tan^{-1} \left(\frac{\alpha_t(i)}{b} \right) \right] \end{aligned} \quad (5.22)$$

Substituting (5.22) into (5.17), we have

$$\hat{b}_{ML} = \operatorname{argmax} \left\{ \sum_{t=1}^T \log \frac{1}{\pi} \left[\tan^{-1} \left(\frac{\beta_t(i)}{b} \right) - \tan^{-1} \left(\frac{\alpha_t(i)}{b} \right) \right] \right\} \quad (5.23)$$

Differentiating (5.23) with respect to b , we obtain

$$\sum_{t=1}^T \frac{\frac{\alpha_t(i)}{b^2 + \alpha_t(i)^2} - \frac{\beta_t(i)}{b^2 + \beta_t(i)^2}}{\tan^{-1} \left(\frac{\beta_t(i)}{b} \right) - \tan^{-1} \left(\frac{\alpha_t(i)}{b} \right)} = 0 \quad (5.24)$$

whose solution can be found by using an iterative root finding algorithm.

In short, our proposed method for enhancing the video reconstructed from multiple compressed copies can be summarized as follows:

Step 1: Estimate the parameters of the Laplacian and Cauchy distributions for each AC coefficient using (5.20) and (5.24), respectively.

Step 2: Obtain the narrow QCS for each integer transform coefficient from the multiple copies using (5.11).

Step 3: Reconstruct each integer transform coefficient as the centroid of the narrow QCS obtained in **Step 2** using (5.14).

Complexity Analysis: It is easy to see that the most computationally intensive part of the proposed method is to construct the narrow QCS and to estimate the model parameter of the distribution for each AC integer transform frequency.

Other than the quantization parameters and quantized values available in the compressed bitstream, the prediction values in the integer transform domain are also needed to compute the narrow QCS, which requires fully decoding every available compressed copy. As only simple and straightforward calculations are required to compute the narrow QCS using (5.10) and (5.11) and the reconstructed integer transform coefficient as the centroid of the narrow QCS using (5.14), this amount of computation is rather insignificant. By applying root finding algorithms such as the Newton-Raphson's method, the model parameter estimation for the distribution of each AC integer transform frequency does not require much computation either in comparison with the whole fully decoding process. Thus, the complexity of the proposed method is approximately equal to the complexity required to decode all available compressed input copies.

In addition, in comparison with the relevant SR or post-processing methods for quantization error reduction, the proposed method generally requires much less computational complexity. Note that the most relevant SR or post-processing methods that share some similarity with the proposed method widely employ the constraint-based techniques with the popular theory of projection onto convex sets (POCS). One of the necessary constraint sets is the smoothness constraint set (SCS) computed in the spatial domain, which also requires fully decoding all the compressed input copies, not to mention the computational load required for the computation of the smoothness criteria. Furthermore, the iterative projection process among various constraint sets requires a number of conversions among the SCS and other constraint sets (e.g., between the spatial domain for the SCS and the transform domain for the QCS), which results in intensively computational load. In order to converge to the optimal solution, a few number of iterations is generally required, which makes the computa-

tional complexity of these methods significantly higher compared with that of the proposed method.

5.4 Video Alignment

In Section 5.3, we propose an effective method to enhance the video reconstructed from multiple compressed copies of the same video content under the assumption that the frames of the available copies are well aligned. However, this assumption may not always hold in practice. For example, the same broadcast video can be encoded by different people starting at slightly different time instances. The same video may also be edited, encoded at different frame rates (e.g., 3-2 pull down), or subjected to frame dropping during the video compression process.

We propose in this section a simple method to align the given compressed video sequences. Without loss of generality, we focus on the alignment of two video sequences here. Let $R = \{r_n : 1 \leq n \leq N_R\}$ and $Q = \{q_n : 1 \leq n \leq N_Q\}$ where r_n and q_n represent the n -th video frames, N_R and N_Q are the total number of frames in the two video sequences. Our objective is to find alignment functions $u(\cdot)$ ($1 \leq u(n) \leq N_R$) and $v(\cdot)$ ($1 \leq v(n) \leq N_Q$) such that frame $r_{u(n)}$ is similar to frame $q_{v(n)}$, for $1 \leq n \leq N_T$, where N_T is the total number of possible matching frame pairs. Mathematically, finding the optimal alignment functions $u(\cdot)$ and $v(\cdot)$ is equivalent to minimizing the matching cost function defined as

$$C(R, Q) = \min_{u,v} \sum_{n=1}^{N_T} w(u(n), v(n)) \cdot d(r_{u(n)}, q_{v(n)}) \quad (5.25)$$

where $d(r_{u(n)}, q_{v(n)})$ is the distance function representing the difference or dissimilarity between frames $r_{u(n)}$ and $q_{v(n)}$, $w(.,.)$ is the weighting function which could place different emphasis on different aligned frame pairs, and $\sum_{n=1}^{N_T} w(u(n), v(n)) = 1$. Frame $r_{u(n)}$ is considered similar to frame $q_{v(n)}$ if their frame distance measure $d(r_{u(n)}, q_{v(n)})$ is sufficiently small. In addition, it should be noted that the minimization is subject to a causal constraint on $u(.)$ and $v(.)$, that is $u(n) \leq u(n+1)$ and $v(n) \leq v(n+1)$ for $1 \leq n \leq N_T$.

It can be seen that the accuracy of the alignment will partly depend on how efficiently the frame distance measure $d(.,.)$ is able to differentiate dissimilar frames. Many sophisticated frame distance measures have been proposed in the literature for image/video matching, such as color histogram, image signatures, etc. Since compressed copies of the same video content exhibit no spatial variations such as different view angles or illumination conditions like the case of existing image/video matching problems, we use here a simple but effective frame distance measure based on the side information extracted from the compressed videos.

Let $S(i)$ be the narrow QCS of the i -th integer transform coefficient obtained from frames $r_{u(n)}$ and $q_{v(n)}$ using (5.11). The proposed frame distance measure between frames $r_{u(n)}$ and $q_{v(n)}$ is defined as

$$d(r_{u(n)}, q_{v(n)}) = \sum_{i=1}^N \mathbf{h}(S(i)) \quad \text{with } \mathbf{h}(S(i)) = \begin{cases} 1 & \text{if } S(i) = \phi \\ 0 & \text{if } S(i) \neq \phi \end{cases} \quad (5.26)$$

where N is the total number of pixels in the video frame. Extensive simulation results show that the distance between aligned frame pairs is generally small compared with that of misaligned frames. To illustrate, we obtained two com-

Figure 5.1: The distance from frame 30 and 50 of one compressed copy to all frames of the other copy of a short video content extracted from a popular situation comedy.

pressed copies of a short video segment from a popular situation comedy by encoding the original movie at different coding parameters. Fig. 5.1 shows the proposed distance measure from a certain frame of one copy (e.g., frame 30 and frame 50) to all frames of the other copy. As can be seen from the figure, the distances between aligned frame pairs are sufficiently small compared with that of misaligned ones. In our work, we define a frame pair to be similar if their distance measure is smaller than some threshold, which is empirically obtained by simulation with a large number of video sequences.

To solve (5.25), we use the forward dynamic programming technique proposed in [170] for video retrieval. Let $C(i, j)$ is the minimum matching cost between two subsequences $R_i = \{r_n : 1 \leq n \leq i\}$ and $Q_j = \{q_n : 1 \leq n \leq j\}$.

Figure 5.2: The optimal path for frame matching between two sequences obtained by using dynamic programming.

The minimum cost $C(i, j)$ for all $1 \leq i \leq N_R$ and $1 \leq j \leq N_Q$ can be computed by using the recursive formula given by

$$C(i, j) = \min \begin{cases} C(i-1, j) & + \frac{1}{N_R+N_Q} \cdot d(r_i, q_j) \\ C(i-1, j-1) & + \frac{2}{N_R+N_Q} \cdot d(r_i, q_j) \\ C(i, j-1) & + \frac{1}{N_R+N_Q} \cdot d(r_i, q_j) \end{cases} \quad (5.27)$$

where $C(i, j) = \infty$ for $i = 0$ or $j = 0$. Hence, the matching frame pairs between video sequences R and Q can be found by determining the optimal path (i.e., the path with minimum final matching cost $C(N_R, N_Q)$). Fig. 5.2 shows the example of such optimal path for frame matching between two sequences. It should be noted that only frame pairs obtained from the optimal path whose the distance measures are smaller than some predefined threshold will be utilized to

enhance the reconstructed video by using the proposed method in Section 5.3.

The complexity for aligning the video sequences R and Q is approximately $O(N_R N_Q)$. To reduce the computational complexity, we can align short segments from the given video sequences. That is, starting from the last aligned frame pair, we obtain a video segment of about 1 to 2 seconds from each sequence. The proposed alignment method is then applied on these two segments under the assumption that they should have at least one aligned frame pair. This assumption is likely to occur in practice if the misalignment of frames is mainly caused by different frame rates or frame dropping during compression. With the consideration of short segments for video alignment, for example by selecting N_R and N_Q less than 30 frames that is about one second of the video content at the frame rate of 30 frames/s, the complexity of the video alignment process can be reduced notably. Furthermore, having considered the multiple copies of the same video content, the proposed video alignment scheme is generally much simpler and has less computational complexity than that of the SR problems with different views, angles, illumination conditions, etc.

To evaluate the performance of the proposed alignment method, we have conducted the experiment on a large number of test sequences. To create the misalignment among the compressed video inputs, the original video sequence was encoded starting from different time instances. Furthermore, we purposely dropped some video frames randomly from the original test sequence before encoding to obtain each compressed copy. The experimental results show that the proposed method can obtain the matching frame pairs among these misaligned compressed copies with a hundred percent of accuracy.

5.5 Analytical Justification

We justify in this section that reconstructing integer transform coefficients using the narrow QCS can generally yield a lower distortion than that of using only the QCS of any single copy.

Let X be a random variable representing an integer transform coefficient, which can be either uniform (for a DC coefficient) or Laplacian/Cauchy (for an AC coefficient). $\hat{X}$ denotes the reconstructed value of X as the centroid of the QCS S . The estimated mean square error $\varepsilon(S)$ can be obtained by (5.13).

Lemma 1: Consider a quantization constraint set $S = [\alpha, \beta]$ and its subset $S' = [\alpha', \beta']$ where $S' \subset S$ (i.e., $\alpha \leq \alpha'$ and $\beta' \leq \beta$). Then for any such subset S' , we have $\varepsilon(S') < \varepsilon(S)$.

Proof: Consider

$$A = \int_{\alpha}^{\beta} x^2 f(x) dx, \quad B = \int_{\alpha}^{\beta} x f(x) dx, \quad \text{and} \quad C = \int_{\alpha}^{\beta} f(x) dx$$

as the functions of β . Then $\varepsilon(S)$ is also a function of β and can be easily obtained as $\varepsilon(S, \beta) = \frac{A}{C} - \frac{B^2}{C^2}$. We first prove that $\varepsilon(S, \beta)$ is an increasing function of β by showing $\frac{\partial \varepsilon(S, \beta)}{\partial \beta} \geq 0$. Rearranging $\varepsilon(S, \beta)$ as $(AC - B^2) \cdot \frac{1}{C^2}$ and taking the derivative of the above function with respect to β , we have

$$\frac{\partial \varepsilon(S, \beta)}{\partial \beta} = (A'C + AC' - 2BB') \frac{1}{C^2} + (AC - B^2) \frac{-2C'}{C^3} \quad (5.28)$$

Using the Leibniz integral rule, it is easy to see that

$$\frac{\partial A}{\partial \beta} = \beta^2 f(\beta), \quad \frac{\partial B}{\partial \beta} = \beta f(\beta), \quad \text{and} \quad \frac{\partial C}{\partial \beta} = f(\beta) \quad (5.29)$$

Replacing (5.29) into (5.28), we obtain

$$\begin{aligned}\frac{\partial \varepsilon(S, \beta)}{\partial \beta} &= (\beta^2 fC + Af - 2\beta fB) \frac{1}{C^2} - 2(AC - B^2) \frac{f}{C^3} \\ &= (\beta^2 C^2 - AC - 2\beta BC + 2B^2) \frac{f}{C^3}\end{aligned}\quad (5.30)$$

Let:

$$I(\beta) = \beta^2 C^2 - AC - 2BC\beta + 2B^2$$

Showing $\frac{\partial \varepsilon(S, \beta)}{\partial \beta} \geq 0$ is equivalent to show $I(\beta) \geq 0$. Taking the first and the second derivatives of $I(\beta)$ with respect to β , we have

$$\begin{aligned}I'(\beta) &= 2\beta C^2 + \beta^2 2Cf - \beta^2 fC - Af - 2(fC\beta^2 + Bf\beta + BC) + 4B\beta f \\ &= 2\beta C^2 - \beta^2 fC - Af - 2BC + 2B\beta f \\ &= 2C(C\beta - B) + f(2B\beta - \beta^2 C - A)\end{aligned}\quad (5.31)$$

$$\begin{aligned}I''(\beta) &= 2f(C\beta - B) + 2C(\beta f + C - \beta f) + f'(2B\beta - \beta^2 C - A) \\ &\quad + f(2B - 2\beta C) \\ &= 2C^2 + f'(2B\beta - \beta^2 C - A)\end{aligned}\quad (5.32)$$

Since f is a symmetric function, we only need to consider $\beta \geq 0$. Let $J(\beta) = 2B\beta - \beta^2 C - A$. It is easy to see that $J'(\beta) = -2(\beta C - B) \leq 0$. Hence $J(\beta)$ is a decreasing function with the increase of β . It follows that $J(\beta) \leq J(\beta = \alpha) = 0$ and $f' \leq 0$, hence $I(\beta)'' \geq 0$. Thus, $I'(\beta)$ increases with β , and $I'(\beta) \geq I'(\beta = \alpha) = 0$. This leads to $I(\beta)$ increasing with β too, thus $I(\beta) \geq I(\beta = \alpha) = 0$. Similarly, we can prove that $\varepsilon(S)$ is a decreasing function of α . Hence, the assertion in Lemma 1 holds. ■

Lemma 1 implies that reconstructing quantized coefficients as the centroid over a narrow QCS can yield a lower distortion on average. Since the proposed method reconstructs the value of each integer transform coefficient by using a narrow QCS that is a subset of the QCSs obtained from the multiple copies, we can reconstruct a video which has a lower distortion, on average, than the video decoded from any given compressed copy. Furthermore, one would expect that more decrease in distortion can be achieved when the size of the narrow QCS decreases. However, how narrow the intersection of the multiple QCSs depends not only on the relation among the sizes of QCSs from multiple compressed copies, which are determined by the corresponding quantization step sizes, but also the position of the QCSs' intervals. In particular, we can unlikely obtain a narrower QCS through intersection when the quantization step sizes are not close to each other. For example, if the quantization step size of one copy is too large compared to another, there is a high probability that the QCS interval determined by the smaller quantization step size is entirely confined by the other QCS interval. In this case, we cannot obtain through intersection a QCS narrower than that of the copy with the smaller quantization step size, resulting in no reduction in distortion compared with that copy. Fig. 5.3(a) illustrates this scenario where the quantization step sizes of Copy 1 and 2 are too large compared with that of Copy 3. This happens when Copy 3 is compressed at much higher quality than the other copies. As a result, we could not obtain a narrower QCS compared with that of Copy 3, which leads to that the quality of the reconstructed video is not better than that of Copy 3. However, it can be seen in Fig. 5.3(b) the relative positions among multiple QCSs can help to reduce the size of the narrow QCS significantly. This is because the position of each independent QCS is partly determined by the prediction value (see (5.7)),

(a) A narrower QCS cannot be obtained when the quantization step sizes of Copy 1 and 2 are too large compared with that of Copy 3.

(b) The relative position of each QCS that is partly determined by the prediction value may help to reduce the size of the narrow QCS significantly.

Figure 5.3: The illustration of how narrow the intersection of the multiple QCSs depends not only on the relation among the sizes of QCSs from multiple compressed copies, but also the position of the QCSs' intervals.

Figure 5.4: Probability of the sizes of the QCS S_1 and the narrow QCS S obtained by the proposed method for different values of Q_2 .

which can be much different for each compressed copy. In addition, one would also expect intuitively the size of the intersected QCS will decrease when more compressed copies are available.

To further illustrate this insight, we provide a simple example. Consider two compressed copies of an integer transform coefficient X , which are coded using two different quantization step sizes Q_1 and Q_2 , respectively. We extensively sampled the values of X based on a Laplacian distribution and quantized with $Q_1 = 14$ and two different values of $Q_2 = 19$ and 53 . As Q_1 is no larger than Q_2 , it is obvious that the coefficient reconstructed from the first copy will have a lower distortion. Fig. 5.4 shows the probability histograms of the sizes of the QCS from the first copy and the narrow QCS obtained through

intersection with different values of Q_2 . As can be seen from the figure, when Q_2 is too large compared to Q_1 (e.g., $Q_2 = 53$), most of the narrow QCSs through intersection have the same size as the QCS from the first copy, resulting in not much distortion reduction. More narrow QCS with smaller sizes compared to that of the first copy can be obtained when Q_2 is close to Q_1 . This explains why using Q_2 of 19 can yield a lower distortion than using Q_2 of 53 (see Fig. 5.4).

5.6 Experimental Results

We have conducted a series of experiments to evaluate the performance of the proposed enhancement method. Our test sequences include ten popular CIF resolution (352×288) sequences, as shown in Table 5.2. These sequences contain different amounts of motion and spatial details, and have been widely tested in the literature of video compression.

Table 5.2: Standard test video sequences

Sequence	No. of frames	Sequence	No. of frames
Flower	300	News	300
Silent	300	Stefan	300
Foreman	300	Tempete	250
Daughter	300	Tennis	150
Coastguard	300	Mobile	300

The experiments were conducted by using the state-of-the-art transform-coding-based video compression standard, namely the H.264/AVC encoder. The multiple copies of the input video were obtained by encoding the same video content using the coding standard with different target bit rates and coding parameters such as the structure of the group-of-pictures (GOP). In what fol-

lows, we shall discuss various scenarios in which the multiple video copies were compressed in different ways, resulting in various possible performance gains.

A. Laplacian and Cauchy Probability Distribution Model

In the first set of the experiments, we evaluate the performance of the proposed method using the Laplacian and Cauchy models, respectively, to resemble the probability distribution of the AC coefficients. We obtained two compressed copies of the Foreman and Stefan sequences by encoding at target bit rates 900 kbits/s and 1000 kbits/s. The GOP of the first copy consists of ten frames with one bidirectional-predictive-coded (B) frame between intra-coded (I) and predictive-coded (P) frames, while the GOP of the second copy consists of twelve frames with two B frames between I and P frames. Fig. 5.5 shows the PSNR results of the best input copy, which is the copy compressed at 1000 kbits/s in this case, and the reconstructed video obtained by the proposed method using the Laplacian and Cauchy models. As can be seen from the figure, the proposed method can consistently reconstruct a video which has a better quality than that of the best input copy. However, the proposed method with the Cauchy model can provide a slightly better reconstructed video quality than that of using the Laplacian model. The superiority of the Cauchy model was also observed on the simulation results of other test sequences. Thus, the Cauchy model is selected to approximate the probability density function of the AC coefficients in our work.

B. Multiple Copies Compressed at Different Target Bit Rates

In the second set of experiments, the same video contents of the test sequences were encoded at different target bit rates. We considered two sets of video input at different bit rate ranges, each consisting of three compressed

Figure 5.5: PSNR results (dB) of the best input copy from the two available compressed copies of the Foreman and Stefan sequences and the reconstructed video obtained by using the proposed method in conjunction with the Laplacian and Cauchy models to approximate the AC coefficient distribution.

Table 5.3: Coding parameters of the standard test video sequences

Video copy no.	Target bitrate (kbits/s)		GOP structure	
	Set 1	Set 2	Case 1	Case 2
1	400	700	M = 1, N = 10	M = 0, N = 9
2	600	900	M = 1, N = 10	M = 1, N = 10
3	800	1000	M = 1, N = 10	M = 2, N = 12

*N is the number of frames in a GOP, M is the number of frames between I & P frames

copies of the same content (see Table 5.3). For comparison purpose, we considered two cases where the video frames from the available copies at the same instance were encoded using the same picture coding types I, P, or B frame (Case 1) or different picture coding types (Case 2). The compressed video copies have different GOP structures as shown in Table 5.3.

Tables 5.4 and 5.5 show the average PSNR results of the video reconstructed from multiple video inputs using the proposed method for Case 1 and Case 2, respectively. Note that the first two copies in each set were used for the case of two video inputs. As expected from the analysis in Section 5.5, the experimental results show that without the original source video or information on the quality of each input video, the proposed method can consistently reconstruct a video which has a better quality (in terms of average PSNR) than that of any input copy. When the input copies are encoded at high-bit rate ranges or more copies are available, the improvement in quality becomes more significant. Specifically, by using all the three copies in Set 2, the video reconstructed by the proposed method can achieve about more than 1.0-dB PSNR improvement than that of the best input copy. In some specific test sequences such as Stefan and Coastguard, the PSNR gain can be more than 2.0 dB (see Table 5.5).

The experimental results also show that the PSNR improvement obtained

Table 5.4: Average PSNR results (in dB) of the best input copy and the video reconstructed by the proposed method from multiple input copies compressed with the same GOP structure at different target bit rates.

Sequence	No. of copies	400-600-800 (kbits/s)			700-900-1000 (kbits/s)		
		Best copy	Proposed	Gain	Best copy	Proposed	Gain
Foreman	2	37.19	37.49	0.30	38.93	39.48	0.55
	3	38.42	38.99	0.57	39.37	40.33	0.96
Mobile	2	22.54	22.86	0.32	23.86	24.28	0.43
	3	23.37	23.84	0.47	24.85	25.48	0.63
Daughter	2	43.93	44.28	0.35	45.23	45.79	0.56
	3	44.87	45.52	0.65	45.58	46.54	0.97
News	2	42.29	42.62	0.33	44.34	44.88	0.54
	3	43.77	44.33	0.56	44.91	45.80	0.89
Silent	2	39.02	39.14	0.12	41.40	41.69	0.29
	3	40.72	41.21	0.49	42.03	43.93	0.90
Flower	2	31.18	31.55	0.38	33.03	33.70	0.66
	3	32.49	33.22	0.72	33.49	34.65	1.16
Stefan	2	30.45	30.73	0.27	32.49	33.05	0.57
	3	31.82	32.43	0.61	32.95	34.08	1.13
Tennis	2	31.45	31.78	0.33	32.75	33.29	0.54
	3	32.40	32.97	0.57	32.89	34.17	1.28
Coastguard	2	32.45	32.80	0.35	34.06	34.65	0.59
	3	33.58	34.24	0.66	34.43	35.66	1.23
Tempete	2	31.52	31.85	0.32	33.23	33.81	0.58
	3	32.74	33.35	0.61	33.59	34.74	1.16

from the set of low-bit rate inputs is lower than that of the high-bit rate set. This can be explained as at low-bit rate range, coarse quantization step sizes are generally used for encoding, resulting in a large QCS for each integer transform coefficient. Furthermore, the QCSs of the low quality copies (e.g., copies 1 and 2 in Set 1) do not contribute much in reducing the size of the narrow QCS obtained by the proposed method. This is because the quantization step sizes used in these copies are generally too large compared to that of the best copy. As a result, the size of the narrow QCS cannot be significantly reduced and hence usually remains the same as that of the best copy. Thus, not much quality

Table 5.5: Average PSNR results (in dB) of the best input copy and the video reconstructed by the proposed method from multiple input copies compressed with different GOP structures at different target bit rates.

Sequence	No. of copies	400-600-800 (kbits/s)			700-900-1000 (kbits/s)		
		Best copy	Proposed	Gain	Best copy	Proposed	Gain
Foreman	2	37.19	37.55	0.36	38.93	39.61	0.68
	3	38.41	39.45	1.04	39.45	41.01	1.56
Mobile	2	22.54	23.03	0.49	23.86	25.26	1.40
	3	25.12	25.93	0.81	26.65	28.17	1.52
Daughter	2	43.93	44.30	0.37	45.23	45.87	0.64
	3	44.87	45.77	0.90	45.59	46.92	1.33
News	2	42.29	42.51	0.22	44.34	44.78	0.44
	3	44.04	44.67	0.63	45.19	46.23	1.04
Silent	2	39.02	39.33	0.31	41.40	41.91	0.51
	3	41.33	41.87	0.54	42.66	43.58	0.92
Flower	2	31.18	31.34	0.16	33.03	33.54	0.51
	3	32.94	33.87	0.93	33.78	35.44	1.66
Stefan	2	30.45	31.14	0.69	32.49	33.70	1.21
	3	31.32	33.01	1.69	32.51	35.02	2.51
Tennis	2	31.45	31.70	0.25	32.75	33.41	0.66
	3	32.62	33.48	0.86	33.14	34.65	1.51
Coastguard	2	32.45	32.88	0.43	34.06	34.89	0.83
	3	33.42	34.58	1.16	34.23	36.20	1.97
Tempete	2	31.52	31.83	0.31	33.23	33.87	0.64
	3	33.03	34.04	1.01	33.82	35.53	1.71

improvement compared to the best copy can be obtained (see the results of Set 1 in Tables 5.4 & 5.5 and discussion in Section 5.5).

Furthermore, we can generally obtain better PSNR gain in the case where the similar frames from the available copies are coded using different picture coding types (Case 2 as shown in Table 5.5) compared with that of using the same picture coding types (Case 1 as shown in Table 5.4). Note that the size of the narrow QCS depends not only on the relation among the sizes of QCSs from multiple compressed copies, but also the relative position of the QCSs' intervals. As explained in Section 5.5, this relative position of each independent

QCS is partly determined by the prediction value, which can be much different when different picture types are used to code similar frames from the available copies. This will help to reduce the size of the narrow QCS obtained by the proposed method significantly, resulting in more distortion reduction.

In addition, it can be seen from Fig. 5.5 that the PSNR gains are also quite consistent and uniformly distributed over the entire sequence. For visual comparison, Figs. 5.6 and 5.7 show the sample frames of the 63th and 77th frames from the Stefan sequence, respectively, which are obtained by the proposed method and by reconstructing from three input copies. The figures show that the proposed method can achieve a better perceived video quality in terms of sharpness and details compared with those reconstructed from the input copies directly. Note that the reconstructed frame from the best input copy, in terms of average PSNR, may not always provide better quality than those reconstructed from other copies as shown in Fig. 5.7. However, the reconstructed frame obtained by the proposed method can still achieve better quality, in terms of both PSNR and visual quality, than the best frame reconstructed from the available input copies (i.e., the reconstructed frame from Copy 2 in this case).

C. Multiple Copies Compressed at the Same Target Bit Rates

In another set of experiments, the input copies were obtained by encoding the test sequences at the same target bit rates. For simplicity, the same GOP structure was used but with different starting frames for different video copies. This is likely to occur in practice, for example, when different people can encode a same broadcast video but starting at slightly different time instances and upload the compressed videos to websites such as YouTube and Google Video. Thus, the encoded picture type (i.e., I, P, or B) for each particular frame may

Copy 1 - Set 2 (33.54 dB)

Copy 2 - Set 2 (34.88 dB)

Copy 3 - Set 2 (36.43 dB)

Proposed method (37.53 dB)

Figure 5.6: Sample frames of the 63th frame from the Stefan sequence obtained by the proposed method and by reconstructing from three input copies.

Copy 1 - Set 2 (31.91 dB)

Copy 2 - Set 2 (33.09 dB)

Copy 3 - Set 2 (29.37 dB)

Proposed method (33.63 dB)

Figure 5.7: Sample frames of the 77th frame from the Stefan sequence obtained by the proposed method and by reconstructing from three input copies.

not be the same among different compressed copies (e.g., it can be an I frame in one copy and a B or P frame in other copies).

Fig. 5.8 shows the PSNR gain of the video reconstructed by the proposed method compared with the best input copy with various target bit rates and number of input copies for the Foreman and Silent sequences. The results show that the proposed method can provide a higher PSNR gain compared with the case in Section 5.6-B. Specifically, processing the Foreman sequence using three copies encoded at bit rates 400 kbits/s, 600 kbits/s, and 800 kbits/s can only obtain 1.04-dB PSNR gain in comparison with the best copy (see Table 5.5). Meanwhile, with three input copies compressed at the same target bit rates of 400 kbits/s and 800 kbits/s, we can yield about 1.49-dB and 1.99-dB PSNR gains, respectively. This is because, unlike the case of different bit rates, the quantization step sizes used to code each copy at the same bit rate are quite close to each other. Furthermore, the same video frame in each copy may be encoded with different picture types, resulting in different motion-compensated prediction values. As the quantization interval of a predicted integer transform coefficient is obtained by adding the integer transform value from the reference frame(s), this could effectively reduce the size of the QCS intersection obtained by the proposed method, leading to a large reduction in the distortion. The experimental results also show that more gain can be achieved with the increase of the bit rates and number of input copies.

D. Multiple Copies Compressed as Variable and Constant Bit Rates

In this set of experiments, we obtained the first compressed copies of the Tennis and Silent sequences by encoding the original video using a constant quantization parameter. The second compressed copy is obtained by encoding at

Figure 5.8: PSNR gain (dB) of the reconstructed video obtained by the proposed method compared with the best input copy with various target bit rates and number of copies for the Foreman and Silent sequences.

the same target bit rate achieved by the first copy. Unlike the case in Section 5.6-C, although both copies have the same target bit rate, the first copy that uses the constant QP for the entire video sequence typically obtains a good performance, in terms of both average PSNR and quality consistency. Fig. 5.9 shows the PSNR results of both available compressed copies and the reconstructed video obtained by using the proposed method. It can be seen that although both copies have different quality in terms of average PSNR, the proposed method can still provide some PSNR gain compared to that of the first copy like the case in Section 5.6-C. The gain obtained by the proposed method is consistent and uniformly distributed over the entire sequence.

D. Application to the Real Video

In the last set of experiments, we evaluated the performance of the proposed method when used together with some real video contents. The real video test sequences were extracted from some featured episodes of a well-known situation comedy with the resolution of 640×480 pixels. The duration of each real video test sequence is about 10 seconds, which consists of about 250 frames. The sample frames of these test sequences are shown in Fig. 5.10. The multiple copies of the input video were obtained by encoding these extracted sequences using the coding standard with different target bit rates and coding parameters, which are shown in Table 5.6.

Table 5.7 shows the average PSNR results of the video reconstructed from multiple video inputs using the proposed method and the best input copy. Like the experiments in Section 5.6-B, the first two copies in each set were used for the case of two video inputs. Similar to the results obtained by using the standard test sequences, we observe that the proposed method can consistently

Figure 5.9: PSNR results (dB) of the two copies of the Tennis and Silent sequences compressed as variable and constant bit rates and the reconstructed video obtained by using the proposed method.

Figure 5.10: Real video test sequences (number of frames in parenthesis).

Table 5.6: Coding parameters of the real video sequences

Video copy no.	Target bitrate (kbits/s)		GOP structure
	Set 1	Set 2	
1	500	900	M = 1, N = 10
2	700	1100	M = 0, N = 9
3	900	1300	M = 2, N = 12

*N is the number of frames in a GOP, M is the number of frames between I & P frames

reconstruct a video which has a better quality than that of the best input copy. Specifically, with the three available compressed copies, the reconstructed video obtained by using the proposed method can obtain about 0.7-dB and 1.2-dB PSNR gains on average for the test sequences in Set 1 and Set 2, respectively. For visual comparison, Figs. 5.11-5.14 show the sample frames from the Soldier sequence, which are obtained by the proposed method and by reconstructing from three input copies. Not surprisingly, the figures show that the proposed method can achieve a better perceived video quality in terms of sharpness and texture details compared with the input copies. The perceptual quality differences can be easily noticed in the regions around the face and the arms of the

Table 5.7: Average PSNR results (in dB) of the best input copy and the video reconstructed by the proposed method from multiple input copies compressed for real video sequences with different GOP structures at different target bit rates.

Sequence	No. of copies	500-700-900 (kbits/s)			900-1100-1300 (kbits/s)		
		Best copy	Proposed	Gain	Best copy	Proposed	Gain
Dancing	2	37.48	38.11	0.63	39.51	40.50	0.99
	3	39.01	39.78	0.77	40.46	41.74	1.28
Superbowl	2	37.23	38.02	0.79	39.38	40.50	1.12
	3	38.96	39.84	0.88	40.40	41.86	1.46
Football	2	36.44	37.11	0.67	38.56	39.65	1.09
	3	38.00	38.85	0.85	39.46	40.86	1.40
London	2	36.89	37.44	0.55	38.67	39.60	0.93
	3	37.67	38.75	1.08	38.97	40.62	1.65
Routine	2	34.68	34.95	0.27	36.80	37.48	0.68
	3	35.50	36.34	0.84	37.21	38.58	1.37
Rugby	2	33.35	33.76	0.41	35.54	36.38	0.84
	3	34.82	35.61	0.79	36.51	37.84	1.33
Soldier	2	36.70	37.38	0.68	38.62	39.78	1.16
	3	38.44	39.24	0.80	40.63	42.43	1.80
Vegas	2	35.73	36.12	0.39	38.01	38.84	0.83
	3	36.86	37.69	0.83	38.57	40.01	1.44

soldier on the left or around the sleeves of the soldier on the right, which are denoted by the rectangular boxes in Figs. 5.11 and 5.13. For better visualization, these regions are enlarged and shown in Figs. 5.12 and 5.14, respectively.

5.7 Summary

We have addressed a new and interesting research problem of blindly enhancing the video reconstructed from multiple compressed video copies of the same video content with different levels of quality. Without making reference to the original source video or information on the quality of the compressed copies, the proposed method effectively exploits the compressed information of different

Figure 5.11: Sample frames of the 20th frame from the Soldier sequence obtained by the proposed method and by reconstructing from three input copies.

Frame 20

Copy 1 (36.34 dB)

Copy 2 (36.74 dB)

Copy 3 (38.66 dB)

Proposed method (39.77 dB)

Figure 5.12: Enlarged regions of the sample frames of the 20th frame from the Soldier sequence obtained by the proposed method and by reconstructing from three input copies for better visualization of the perceptual differences.

Figure 5.13: Sample frames of the 64th frame from the Soldier sequence obtained by the proposed method and by reconstructing from three input copies.

Frame 64

Copy 1 (36.21 dB)

Copy 2 (35.69 dB)

Copy 3 (36.50 dB)

Proposed method (38.49 dB)

Figure 5.14: Enlarged regions of the sample frames of the 64th frame from the Soldier sequence obtained by the proposed method and by reconstructing from three input copies for better visualization of the perceptual differences.

video copies to reconstruct a video that has a better quality in terms of PSNR than the best compressed copy. Specifically, each coefficient of the reconstructed in the transform domain is estimated using a narrow quantization constraint set obtained from the multiple compressed copies together with a Laplacian or Cauchy distribution model for each AC frequency to minimize the distortion. Analytical and experimental results show that the video reconstructed by the proposed method not only yields a lower distortion than any given compressed copy but also achieves a significant PSNR gain compared to the best copy. Furthermore, a similar approach can be easily extended to other transform-based coding schemes such as DCT-based or wavelet-based transform coding.

Chapter 6

Conclusion and Future Work

In this chapter, we present the concluding remarks and summarize the main contributions of the thesis. We also provide several examples of potential applications of interest, where various techniques developed in this thesis can be integrated together cohesively to deliver better video quality. The discussion of possible research directions for the future work is given at the end of this chapter.

6.1 Conclusion

In this thesis, we have focused on optimizing video encoder and transcoder to achieve better performance in terms of complexity and quality. We have proposed a variety of efficient schemes to reduce the complexity for video encoder and transcoder while preserving the visual quality. Furthermore, we have provided effective post-processing techniques to address the issue of quality degradation, which is influenced by these coding and transcoding solutions. We

summarize the key developments and contributions of the thesis as follows:

- **Fast block-matching algorithm for motion estimation in video encoder**

The conventional video compression algorithms adopted in many standards can achieve high coding efficiency but the process is computationally costly and may be not suitable for many applications. We have studied and focused on the motion estimation and compensation process in a video encoder, which is arguably one of the most computationally intensive parts, to reduce the computational complexity. Of many fast block-matching motion estimation techniques, we have mainly focused on the simplified block-matching (BM) measures.

More specifically, we have proposed in this thesis new BM measures for complexity reduction with acceptable visual quality preservation. The proposed measures evaluate the match between two blocks based on block features such as block sum and block variance. The concept of integral frame attributes was introduced for fast computation of these block features. To further speed up the matching process, we have integrated the proposed BM measures in a two-step approach and a partial summation elimination scheme.

We have performed the analytical study and a large number of experiments to show the effectiveness of the proposed method. In particular, compared with the conventional SAD measure, the proposed block-matching measures can reduce the number of arithmetic operations notably, when used in conjunction with the FS and TSS algorithms. This saving in computation is gained without degrading much the compressed video quality.

Furthermore, the number of computations can be further reduced by using the two-step approach or the partial summation elimination scheme to reject impossible candidate blocks in the early stage of the search. The experiment results also showed that the proposed measures could be used in many existing fast search algorithms besides the TSS algorithm.

- **Efficient scheme for syntax and downsizing transcoding with enhanced rate control**

Video transcoding is crucial in supporting universal access for better multimedia communications and services among today's heterogeneous networks and devices. Similar to video compression, the conventional transcoding approach by cascaded decoder and encoder requires a large amount of processing time and computing resource, and is unable to meet the limited resources in real-time applications. We have examined many transcoding architectures and techniques to reduce the complexity and improve the quality of the transcoding process. We have mainly focused on the cascaded pixel-domain transcoder for syntax and spatial resolution reduction transcoding by using the useful side information in the precoded video to speed up the transcoding process.

In our work, we have addressed the transcoding issues between the existing standard H.263 and the latest H.264/AVC standard. Due to some advanced coding modes in H.264/AVC, besides a fast motion vector re-estimation process, efficient methods for other computationally intensive parts such as intra-prediction mode selection or rate distortion optimization for mode selection have been presented. In particular, a vector median filter together with motion vector refinement have been proposed to re-

estimate the required motion vectors for the transcoded video from those in the precoded video. To reduce the complexity of intra-prediction mode selection, we have proposed to reduce the number of possible intra modes based on the dominant edge in the block, which can be obtained from the integer transform domain.

In addition, to improve the quality of the transcoded video, we have proposed an enhanced rate control method for H.264/AVC. The proposed method is based on the key observation of the bit rate relationship between the precoded and transcoded videos, which can be approximated by a quadratic function of the quantization step size ratio. The proposed rate control method used such quadratic model for selecting quantization parameters at the sequence and frame levels together with a new frame-layer bit allocation scheme based on the frame complexity measured from the side information in the precoded video.

We have conducted the experimental results and shown the accuracy of the model and the effectiveness of the proposed methods. In particular, the PSNR obtained by the proposed methods is only a little inferior to that obtained by the cascaded H.264/AVC recoding scheme, while the total transcoding time can be reduced by a factor of 6. Furthermore, the proposed rate control method could meet the target bit rate more accurately and provide more consistent video quality compared with that of existing H.264/AVC rate control scheme.

- **Blind video enhancement for multiple compressed copies**

Although being able to reduce the complexity efficiently, fast compression and transcoding schemes often affect the perceptual quality. In this thesis,

we have mainly concerned the issue of visual quality degradation, which is due to the quantization noise introduced in block-based video encoder and transcoder. We have mostly focused on the post-processing techniques to improve the quality of the reconstructed video from the video compression and transcoding processes.

Specifically, we have addressed a new and interesting research problem of blindly enhancing the video reconstructed from multiple compressed video copies of the same video content with different levels of quality. We have proposed an efficient method to reconstruct the video that has a better quality in terms of average PSNR than any of the available copies. The proposed method is considered “blind” due to it not making reference to the original source video or information on the quality of the compressed copies.

In particular, each coefficient of the reconstructed video is estimated using a narrow quantization constraint set obtained from multiple compressed copies in transform domain to minimize the distortion. Instead of assuming uniform distribution, we have proposed to approximate the distribution for each AC frequency by Laplacian and Cauchy models, in which the model parameters can be well estimated using the side information from the compressed input copies. Analytical and experimental results have shown that the video reconstructed by the proposed method not only yields a lower distortion than any given compressed copy, but also achieves a significant PSNR gain compared to the best copy. Furthermore, a similar approach can be easily extended to other transform-based coding schemes such as DCT-based or wavelet-based transform coding.

In summary, we have proposed in this thesis various schemes for improving video encoder and transcoder performances. These schemes have focused on the related areas, which ensure for better and faster visual communications services and multimedia applications in terms of efficient storage, transmission, and display. The improved techniques for video encoder and transcoder allow for efficient storage and transmission among today's heterogeneous networks and devices. Meanwhile, the optimized post-processing techniques complement the inadequacy in terms of quality degradation of the decoded or transcoded video for better delivered video quality.

We present a figure (Fig. 6.1) to illustrate a potential application, where these proposed schemes can be integrated together cohesively to deliver better media quality. In this application example, users may capture or extract video contents from various sources such as video camcorder, DVD content, or live broadcasting program, which utilize compression techniques to reduce the data rate before sharing or transmitting over the network. Here, attaining the high video quality may not be the main objective but compression with low complexity is mostly concerned. For example, one may want to capture and stream the live broadcasting program over the Internet; or the live video content in a surveillance system needs to be transmitted to the monitoring rooms in real time. In another example, users may prefer efficient compression methods to extract the high quality video contents (e.g., DVD, high-definition camcorder) with a reasonable lower quality to share over the Internet due to the limited processing time or transmission bandwidth. Thus, by employing the fast coding schemes like the one proposed in Chapter 3, one can meet such constraints and requirements.

Figure 6.1: Potential application of interest for multimedia applications and services.

In addition, the compressed videos distributed over the network are not necessarily accessible by different users with diversified devices and channel profiles. For example, the streaming video compressed by one coding standard may not be processed by users with the available decoder compliant to a different standard. Also users with advanced 3G mobiles may be not able to process these compressed videos due to the limited bandwidth (e.g., 802.11b or ISDN gateway), small display screen, or limited processing power. To compound this inadequacy, the needs for efficient video transcoding schemes like those proposed in Chapter 4 is necessary to adapt compressed videos to such new application constraints.

Furthermore, while efficient compression and transcoding solutions are necessary to meet the scarce resources and constraints at the expense of quality degradation, attaining better visual quality of these compressed videos may be another main interest in some scenario. For example, one may experience more

than one compressed copy of the same video content (e.g., movies or video clips extracted from DVD) with different levels of visual quality. The quality differences are due to the fact that these copies are generated and distributed by different users, which may apply different coding parameters and methods to meet their own constraints or preferences. In this situation, it may be desirable to select and keep a copy with the best quality if the resources and constraints are no longer concerned. As the original source video is not always available, how to choose or derive a video of the best quality from these copies is non-trivial as we don't definitively know which of the multiple copies, which frame of a copy, and which region of a frame have the best quality. Therefore, the blind video enhancement method proposed in Chapter 5 can be exploited here to reconstruct a video with a better quality.

6.2 Future Work

We have shown the work in this thesis provides substantial contributions to understanding and improvement of video encoder and transcoder for a better delivery of multimedia applications. Still, there are some issues that go beyond the scope of this thesis and serve as motivating future research directions.

With the fast-growing development of networks and the emerging new generation mobile phones, there are high demands for efficient techniques to deliver quality media information in various mobile applications for mass users. Video coding and transcoding solutions have still played an important role to customize the multimedia for specific requirements, but it is more challenging here as fine adaptation to the user's needs is generally not possible in one-to-many

scenarios, and compromises have to be derived to offer an acceptably good overall performance for mass mobile users. To maximize the end-user experience and optimize the system and network resources for mobile media applications, complexity-aware, content-aware, and attention-based media adaptation techniques have attracted much research interest and should be explored in the future. Furthermore, with the widespread deployment of wireless networks in mobile media applications, transmitting data over these existing error-prone networks can be very unreliable due to time-varying interference and burst channel errors, and pose many challenges to streaming rich-media information, especially mobile video. Thus, transcoding techniques that adapt video for better error resilience should also be considered for better delivered media quality.

In the interesting problem of video enhancement from multiple compressed copies, the proposed method can provide a better reconstructed video; however there are cases where it falls short and should be addressed in the future. One of examples is when the compressed input copies have different temporal resolutions, the number of input frames available for enhancement at different time instances will vary. Certain time instances may involve more high-quality input frames while the others may not. This happens when one input copy is encoded at high quality with different frame rate compared with other copies. Although the reconstructed video obtained by the proposed method can still achieve better average PSNR than any of the available copies, the consistency of the visual quality may not be guaranteed. To alleviate the problem, the reconstruction of enhanced frames at a certain time instance should exploit the information not only from multiple compressed copies, but also from the neighboring enhanced frames to achieve smooth visual quality.

In addition, we have considered only compressed copies obtained by using the same video coding standard without any spatial variation in the current work. We expect to extend the proposed solutions by taking into account different coding standards with different features (e.g., 8×8 DCT transform in H.263 and MPEG-2 instead of 4×4 integer transform in H.264/AVC) or different spatial resolutions. With considering these variations, obtaining the narrow quantization constraint set may be no longer straightforward and becomes challenging. To address these issues, one possible solution for the future work is to reconstruct the enhanced video based on the theory of projection into convex set. By determining the optimization function to improve the visual quality under constraint sets derived from multiple compressed copies, we can overcome this mismatch problem due to various coding features and modifications.

Although we have mentioned several applications of the blind video enhancement approach, we recommend to extend it to other potential applications. One of the application scenarios of interest is scalable video coding, which has been developed and standardized recently. In scalable video coding, the video is compressed into one or more layers of different quality and spatial resolutions. The decoded frames of these layers can be considered as the multiple compressed copies of the same visual content. Instead of using the previously decoded frame in one layer as the reference frame for motion estimation and prediction, we can exploit the proposed enhancement method to reconstruct a reference frame with better quality. The enhanced reference frame is reconstructed from the multiple decoded copies of the same frame from various layers of different quality and resolutions. With the enhanced reference frame, motion-compensated prediction can be obtained more accurately, which can likely lead to a higher coding efficiency in scalable video coding.

Last but not the least, the newly proposed video enhancement problem of multiple copies also shares some similarity with multiple description coding. By considering each compressed copy as a side description, it can be considered as finding the optimal central description with all side descriptions available at the decoder side. Intuitively, theory and techniques of multiple description coding may be applied here. Although this may be not straightforward as multiple description coding controls the correlation of side descriptions to obtain the optimal central description subject to distortion constraints. Meanwhile, the proposed problem does not have any control on how the side descriptions are obtained. Thus, it may be complicated to design an optimal central decoder like in the case of multiple description coding as we do not have the control or information at the encoder side. Nevertheless, this is really an interesting direction which may open more efficient solutions for this newly proposed research problem and should be explored in the future.

Publications

1. Viet-Anh Nguyen, Yap-Peng Tan, and Zhenzhong Chen, "Blind video enhancement from multiple compressed copies," *to be submitted to Signal Processing: Image Commun.*, Dec. 2008.
2. Viet-Anh Nguyen and Yap-Peng Tan, "Efficient block-matching motion estimation based on integral frame attributes," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 3, pp. 375-385, Mar. 2006.
3. Viet-Anh Nguyen and Yap-Peng Tan, "Efficient video transcoding from H.263 to H.264/AVC standard with enhanced rate control," *EURASIP Journal on Applied Signal Processing*, no. 1, pp. 1-15, Jan. 2006.
4. Viet-Anh Nguyen and Yap-Peng Tan, "Fast block-based motion estimation using integral frames," *IEEE Signal Process. Letters*, vol. 11, no. 9, pp. 744-747, Sept. 2004.
5. Viet-Anh Nguyen, Yap-Peng Tan, and Zhenzhong Chen, "Efficient multi-copy video enhancement technique in transform domain," *to be submitted to IEEE Int. Conf. Image Process.*, Jan. 2009.
6. Viet-Anh Nguyen, Yap-Peng Tan, and Weisi Lin, "Adaptive downsampling/upsampling for better video compression at low bit rate," *in Proc. IEEE Int. Symp. Circuits and Systems*, pp. 1624-1627, May 2008.
7. Viet-Anh Nguyen and Yap-Peng Tan, "Efficient H.263 to H.264/AVC video transcoding using enhanced rate control," *in Proc. IEEE Int. Conf. Image Process.*, no. 3, pp. 652-644, Sept. 2005.
8. Viet-Anh Nguyen and Yap-Peng Tan, "Efficient video transcoding between H.263 and H.264/AVC standards," *in Proc. IEEE Int. Symp. Circuits and Systems*, no. 2, pp. 904-907, May 2005.

9. You-Neng Xiao, Hong Lu, Xiangyang Xue, Viet-Anh Nguyen, and Yap-Peng Tan, "Efficient rate control for MPEG-2 to H.264/AVC transcoding," in *Proc. IEEE Int. Symp. Circuits and Systems*, no. 2, pp. 1238-1241, May 2005.
10. Viet-Anh Nguyen and Yap-Peng Tan, "Block matching using integral frame attributes," in *Proc. IEEE Pacific-Rim Conf. Multimedia*, no. 3, pp. 175-183, Dec. 2004.

Bibliography

- [1] H. G. Musmann, P. Pirsch, and H. J. Grallert, "Advances in picture coding," in *Proc. of IEEE*, vol. 73, no. 4, pp. 523–548, Apr. 1985.
- [2] A. K. Jain, *Fundamentals of digital image processing*. Information and System Sciences, Prentice Hall, 1989.
- [3] K. Sayood, *Introduction to data compression*. Multimedia and Information Systems, Morgan Kaufmann Publishers Inc., 2000.
- [4] D. A. Huffman, "A method for the construction of minimum-redundancy codes," in *Proc. of IRE*, vol. 40, no. 9, pp. 1098–1101, Sept. 1952.
- [5] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Communications of the ACM*, vol. 30, no. 6, pp. 520–540, Jun. 1987.
- [6] ISO/IEC JTC1/SC29/WG11, "Information technology - Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbits/s," *Part 2: Video. Draft ISO/IEC 11172-2 (MPEG-1)*, ISO/IEC, Geneva, 1991.
- [7] ISO/IEC JTC1/SC29/WG11, "Information technology - Generic coding of moving pictures and associated audio," *Part 2: Video. Draft ISO/IEC 13818-2 (MPEG-2) and ITU-T Recommendation H.262*, ISO/IEC and ITU-T, Geneva, 1994.
- [8] MPEG Video Group, "Report of ad-hoc group on the evaluation of tools for nontested functionalities of video submissions to MPEG-4," *Munich meeting, document ISO/IEC/JTC1/SC29/WG11 N0679*, Jan. 1996.
- [9] CCITT H.261, "Video codec for audiovisual services at $p \times 64$ kbit/s," 1990.

- [10] ITU-T/SG15, "Video coding for low bitrate communication," *ITU-T Recommendation H.263, Version 1*, ITU-T, Geneva, 1996.
- [11] G. J. Sullivan, T. Wiegand, and T. Stockhammer, "Draft H.26L video coding standard for mobile applications," in *Proc. IEEE Int. Conf. Image Process.*, vol. 3, pp. 573–576, Oct. 2001.
- [12] R. Dugad and N. Ahuja, "A scheme for spatial scalability using non-scalable encoders," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 10, pp. 993–999, Oct. 2003.
- [13] J. Li, K. Yu, T. He, Y. Lin, S. Li, and Y. Q. Zhang, "Scalable portrait video for mobile video communication," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 5, pp. 376–384, May 2003.
- [14] J. L. Lin, W. L. Hwang, and S. C. Pei, "SNR scalability based on bitplane coding of matching pursuit atoms at low bit rates: fine-grained and two-layer," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 1, pp. 3–14, Jan. 2005.
- [15] S. E. Kim, J. K. Han, and J. G. Kim, "Efficient motion estimation algorithm for MPEG-4 to H.264 transcoder," in *Proc. IEEE Int. Conf. Image Process.*, vol. 3, pp. 656–659, Sept. 2005.
- [16] X. Xiu, L. Zhuo, and L. Shen, "A H.264 bit rate transcoding scheme based on PID controller," in *Proc. IEEE Int. Symp. Commun. Inform. Technol.*, vol. 2, pp. 1074–1077, Oct. 2005.
- [17] Y. P. Tan and H. Sun, "Fast motion re-estimation for arbitrary downsizing video transcoding using H.264/AVC standard," *IEEE Trans. Consumer Electron.*, vol. 50, no. 3, pp. 887–894, Aug. 2004.
- [18] P. Peter, D. Nicolas, and G. Winfried, "VLSI architectures for video compression - A survey," in *Proc. of IEEE*, vol. 83, no. 2, pp. 220–246, Feb. 1995.
- [19] Y. Kim, C. S. Park, and S. J. Ko, "Frequency domain post-processing technique based on POCS," *IEEE Electron. Letters*, vol. 39, no. 22, pp. 1583–1584, Oct. 2003.
- [20] Y. Yang and L. Boroczky, "A new enhancement method for digital video applications," *IEEE Trans. Consumer Electron.*, vol. 48, no. 3, pp. 435–443, Aug. 2002.

- [21] S. Liu and A. C. Bovik, "Efficient DCT-domain blind measurement and reduction of blocking artifacts," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 12, pp. 1139–1149, Dec. 2002.
- [22] Y. Luo and R. K. Ward, "Removing the blocking artifacts of block-based DCT compressed images," *IEEE Trans. Image Process.*, vol. 12, no. 7, pp. 838–842, Jul. 2003.
- [23] Y. Wang, J. Ostermann, and Y. Q. Zhang, *Video processing and communications*. Signal Processing, Prentice Hall, Sept. 2001.
- [24] ITU-R BT.601-5, "Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios," tech. rep., formerly CCIR 601, 1998.
- [25] EBU Technical 3229, "High definition (HD) image formats for television production," tech. rep., 2004.
- [26] E. Adachi and S. Horiguchi, "Multi-resolutional optical flow estimation with local optimization," in *Proc. IEEE Int. Conf. Image Process.*, vol. 2, pp. 257–260, Sept. 2002.
- [27] H. M. Hang, Y. M. Chou, and S. C. Cheng, "Motion estimation for video coding standards," *Journal of VLSI Signal Process. Syst.*, vol. 17, no. 2, pp. 113–136, Nov. 1997.
- [28] Y. Wang and O. Lee, "Use of 2-D deformable mesh structures for video compression, Part 1 - the synthesis problem: Mesh based function approximation and mapping," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 6, pp. 636–646, Dec. 1996.
- [29] B. Girod, "Motion-compensating prediction with fractional-pel accuracy," *IEEE Trans. Commun.*, vol. 41, no. 21, pp. 604–612, 1993.
- [30] "Coding of audio-visual objects - Part 2: Visual," *ISO/IEC 14 496-2 (MPEG-4 Visual Version 1)*, Apr. 1999.
- [31] T. Koga, K. Inuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion-compensated interframe coding for video conferencing," in *Proc. of National Telecomm. Conf.*, pp. G5.3.1–G5.3.5, Nov. 1981.
- [32] R. Li, B. Zeng, and M. L. Liou, "A new three-step search algorithm for block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, no. 4, pp. 438–442, Aug. 1994.

- [33] L. M. Po and W. C. Ma, "A novel four-step search algorithm for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 3, pp. 313–317, Jun. 1996.
- [34] L. K. Liu and E. Feig, "A block-based gradient descent search algorithm for block motion estimation in video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 4, pp. 419–422, Aug. 1996.
- [35] S. Zhu and K. K. Ma, "A new diamond search algorithm for fast block matching motion estimation," *IEEE Trans. Image Process.*, vol. 9, no. 2, pp. 287–290, Feb. 2000.
- [36] H. So, J. Kim, W. K. Cho, and Y. S. Kim, "Fast motion estimation using modified diamond search patterns," *IEEE Electron. Letters*, vol. 41, no. 2, pp. 62–63, Jan. 2005.
- [37] C. H. Cheung and L. M. Po, "A novel cross-diamond search algorithm for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 12, pp. 1168–1177, Dec. 2002.
- [38] C. Zhu, X. Lin, and L. P. Chau, "Hexagon-based search pattern for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 5, pp. 349–355, May 2002.
- [39] B. Liu and A. Zaccarin, "New fast algorithms for the estimation of block motion vectors," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, no. 2, pp. 148–157, Apr. 1993.
- [40] J. Y. Nam, J. S. Seo, J. S. Kwak, M. H. Lee, and Y. H. Ha, "New fast-search algorithm for block matching motion estimation using temporal and spatial correlation of motion vector," *IEEE Trans. Consumer Electron.*, vol. 46, no. 4, pp. 934–942, Nov. 2000.
- [41] P. I. Hosur and K. K. Ma, "Motion vector field adaptive fast motion estimation," in *Proc. IEEE Int. Conf. Inform. Commun. Signal Process.*, Dec. 1999.
- [42] A. M. Tourapis, O. C. Au, and M. L. Liou, "Predictive motion vector field adaptive search technique (PMVFAST) - Enhancing block based motion estimation," in *Proc. Visual Commun. Image Process.*, Jan. 2001.
- [43] H. Gharavi and M. Mills, "Block matching motion estimation algorithms: new results," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 37, no. 5, pp. 649–651, 1990.

- [44] H. O. Yong, S. Erturk, and T. G. Chang, "Low complexity video encoding with one-bit transform based network-driven motion estimation," *IEEE Trans. Consumer Electron.*, vol. 53, no. 2, pp. 601–608, May 2007.
- [45] X. Song, T. Chiang, and Y. Q. Zhang, "A scalable hierarchical motion estimation algorithm for MPEG-2," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 4, pp. 126–129, Jun. 1998.
- [46] H. Chung, D. Romacho, and A. Ortega, "Fast long-term motion estimation for H.264 using multiresolution search," in *Proc. IEEE Int. Conf. Image Process.*, vol. 1, pp. 905–908, Sept. 2003.
- [47] J. Zan, M. O. Ahmad, and M. N. Swamy, "A multiresolution motion estimation technique with indexing," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 2, pp. 157–165, Feb. 2006.
- [48] X. Song, T. Chiang, X. Lee, and Y. Q. Zhang, "New fast binary pyramid motion estimation for MPEG2 and HDTV encoding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 7, pp. 1015–1028, Oct. 2000.
- [49] W. Li and E. Salari, "Successive elimination algorithm for motion estimation," *IEEE Trans. Image Process.*, vol. 4, no. 1, pp. 105–107, Jan. 1995.
- [50] X. Q. Gao, C. J. Duanmu, and C. R. Zou, "A multilevel successive elimination algorithm for block matching motion estimation," *IEEE Trans. Image Process.*, vol. 9, no. 3, pp. 501–504, Mar. 2000.
- [51] J. N. Kim, D. K. Kang, S. C. Byun, I. L. Lee, and B. H. Ahn, "A fast full search motion estimation algorithm using sequential rejection of candidates from hierarchical decision structure," *IEEE Trans. Broadcasting*, vol. 48, no. 1, pp. 43–46, Mar. 2002.
- [52] S. W. Liu, S. D. Wei, and S. H. Lai, "Fast optimal motion estimation based on gradient-based adaptive multilevel successive elimination," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 2, pp. 263–267, Feb. 2008.
- [53] C. K. Cheung and L. M. Po, "Adjustable partial distortion search algorithm for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 1, pp. 100–110, Jan. 2003.
- [54] X. Jing and L. P. Chau, "Partial distortion search algorithm using predictive search area for fast full-search motion estimation," *IEEE Signal Process. Letters*, vol. 14, no. 11, pp. 840–843, Nov. 2007.

- [55] H. Sun, W. Kwok, and J. W. Zdepski, "Architectures for MPEG compressed bitstream scaling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 2, pp. 191–199, Apr. 1996.
- [56] P. Yin, M. Wu, and B. Liu, "Video transcoding by reducing spatial resolution," in *Proc. IEEE Int. Conf. Image Process.*, vol. 1, pp. 972–975, 2000.
- [57] T. Shanableh and M. Ghanbari, "Heterogeneous video transcoding to lower spatial-temporal resolutions and different encoding formats," *IEEE Trans. Multimedia*, vol. 2, no. 2, pp. 101–110, Jun. 2000.
- [58] N. Roma and L. Sousa, "Insertion of irregular-shaped logos in the compressed DCT domain," in *Proc. 14th Int. Conf. Digital Signal Process.*, vol. 1, pp. 125–128, 2002.
- [59] G. D. L. Reyes, A. R. Reibman, S. F. Chang, and J. C. I. Chuang, "Error-resilience transcoding for video over wireless channels," *IEEE J. Select. Areas Commun.*, vol. 18, no. 6, pp. 1063–1074, Jun. 2000.
- [60] A. Eleftheriadis and D. Anastassiou, "Constrained and general dynamic rate shaping of compressed digital video," in *Proc. IEEE Int. Conf. Image Process.*, vol. 3, pp. 396–399, Oct. 1995.
- [61] J. Youn and M. T. Sun, "Video transcoding with H.263 bit stream," *J. Visual Commun. Image Represent.*, vol. 11, no. 4, pp. 385–403, Dec. 2000.
- [62] P. A. A. Assuncao and M. Ghanbari, "A frequency domain video transcoder for dynamic bit rate reduction of MPEG-2 bit streams," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 8, pp. 953–967, Dec. 1998.
- [63] J. Xin, C. W. Lin, and M. T. Sun, "Digital video transcoding," in *Proc. of IEEE*, vol. 93, no. 1, pp. 84–97, Jan. 2005.
- [64] P. Yin, A. Vetro, B. Liu, and H. Sun, "Drift compensation for reduced spatial resolution transcoding: a summary," *IEEE Circuits and Systems Magazine*, vol. 4, no. 2, pp. 32–36, 2004.
- [65] S. F. Chang and D. G. Messerschmitt, "Manipulation and composition of MC-DCT compressed video," *IEEE J. Select. Areas Commun.*, vol. 13, no. 1, pp. 1–11, Jan. 1995.

- [66] S. Z. Liu and A. C. Bovik, "Local bandwidth constrained fast inverse motion compensation for DCT-domain video transcoding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 5, pp. 309–319, May 2002.
- [67] W. C. Siu, K. T. Fung, and Y. L. Chan, "A compressed-domain heterogeneous video transcoder," in *Proc. IEEE Int. Conf. Image Process.*, vol. 4, pp. 2761–2764, Oct. 2004.
- [68] H. Kato, Y. Nakajima, and T. Sano, "Integrated compressed domain resolution conversion with deinterlacing for DV to MPEG-4 transcoding," in *Proc. IEEE Int. Conf. Image Process.*, vol. 4, pp. 2809–2812, Oct. 2004.
- [69] W. Zhu, K. H. Yang, and M. J. Beack, "CIF-to-QCIF video bitstream down-conversion in the DCT domain," *Bell Labs Tech. J.*, vol. 3, no. 3, pp. 21–29, 1998.
- [70] N. Merhav, "Multiplication-free approximate algorithms for compressed-domain linear operations on images," *IEEE Trans. Image Process.*, vol. 8, no. 2, pp. 247–254, Feb. 1999.
- [71] U. V. Koc and K. J. R. Liu, "Discrete-cosine/sine-transform based motion estimation," in *Proc. IEEE Int. Conf. Image Process.*, vol. 3, pp. 771–775, Nov. 1994.
- [72] B. Shen, I. K. Sethi, and B. Vasudev, "Adaptive motion-vector resampling for compressed video downscaling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 6, pp. 929–936, Sept. 1999.
- [73] J. N. Hwang and T. D. Wu, "Motion vector re-estimation and dynamic frame-skipping for video transcoding," *Conference Record of the Thirty-Second Asilomar on Signals, Systems and Computers*, vol. 2, pp. 1606–1610, Nov. 1998.
- [74] K. D. Seo and J. K. Kim, "Motion vector refinement for video downsampling in the DCT domain," *IEEE Signal Process. Letters*, vol. 9, no. 11, pp. 356–359, Nov. 2002.
- [75] Y. Liang, L. P. Chau, and Y. P. Tan, "Arbitrary downsizing video transcoding using fast motion vector re-estimation," *IEEE Signal Process. Letters*, vol. 9, no. 11, pp. 352–355, Nov. 2002.
- [76] J. Youn, M. T. Sun, and C. W. Lin, "Motion vector refinement for high-performance transcoding," *IEEE Trans. Multimedia*, vol. 1, no. 1, pp. 30–40, Mar. 1999.

- [77] J. Nakajima, H. Tsuji, Y. Yashima, and N. Kobayashi, "Motion vector re-estimation for fast video transcoding from MPEG-2 to MPEG-4," in *Proc. of Workshop and Exhibition on MPEG-4*, pp. 87–90, Jun. 2001.
- [78] N. Bjork and C. Christopoulos, "Transcoder architecture for video coding," *IEEE Trans. Consumer Electron.*, vol. 44, no. 1, pp. 88–98, 1998.
- [79] G. Shen, B. Zeng, Y. Q. Zhang, and M. L. Liou, "Transcoder with arbitrarily resizing capability," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 5, pp. 25–28, 2001.
- [80] J. Xin, M. T. Sun, K. Chun, and B. S. Choi, "Motion re-estimation for HDTV to SDTV transcoding," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 4, pp. 715–718, 2002.
- [81] M. J. Chen, M. C. Chu, and C. W. Pan, "Efficient motion-estimation algorithm for reduced frame-rate video transcoder," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 4, pp. 269–275, Apr. 2002.
- [82] H. Sun, Y. P. Tan, and Y. Liang, "Fast motion vector and bitrate re-estimation for arbitrary downsizing video transcoding," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 2, pp. 856–859, 2003.
- [83] K. D. Seo and J. K. Kim, "Fast motion vector re-estimation for transcoding MPEG-1 into MPEG-4 with lower spatial resolution in DCT-domain," *Signal Processing: Image Commun.*, vol. 19, no. 4, pp. 299–312, 2004.
- [84] K. A. Lan and J. N. Hwang, "Scene context dependent reference frame placement for MPEG video coding," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, vol. 4, pp. 2997–3000, Apr. 1997.
- [85] J. N. Hwang, T. D. Wu, and C. W. Lin, "Dynamic frame-skipping in video transcoding," in *Proc. IEEE Second Workshop Multimedia Signal Process.*, pp. 616–621, Dec. 1998.
- [86] Y. P. Tan, H. Sun, and Y. Liang, "On the methods and applications of arbitrary downsizing video transcoding," in *Proc. IEEE Int. Conf. Multimedia and Expo*, vol. 1, pp. 609–612, Aug. 2002.
- [87] I. Ahmad, X. Wei, Y. Sun, and Y. Q. Zhang, "Video transcoding: an overview of various techniques and research issues," *IEEE Trans. Multimedia*, vol. 7, no. 5, pp. 793–804, Oct. 2005.

- [88] H. C. Reeves and J. S. Lim, "Reduction of blocking effect in image coding," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, vol. 8, pp. 1212–1215, Apr. 1983.
- [89] A. W. C. Liew and Y. Hong, "Blocking artifacts suppression in block-coded images using overcomplete wavelet representation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 4, pp. 450–461, Apr. 2004.
- [90] M. Liou, "Overview of the p×64 kbit/s video coding standard," *Communications of the ACM*, vol. 34, no. 4, pp. 59–63, Apr. 1991.
- [91] K. K. Kwon, S. H. Im, and D. S. Lim, "Deblocking algorithm in MPEG-4 video coding using block boundary characteristics and adaptive filtering," in *Proc. IEEE Int. Conf. Image Process.*, vol. 3, pp. 541–544, Sept. 2005.
- [92] H. S. Kong, Y. Nie, A. Vetro, H. Sun, and K. E. Barner, "Coding artifacts reduction using edge map guided adaptive and fuzzy filtering," in *Proc. IEEE Int. Conf. Multimedia and Expo*, vol. 2, pp. 1135–1138, Jun. 2004.
- [93] S. Kim and J. Jeong, "Enhancement of wavelet-coded images via novel directional filtering," in *Proc. IEEE Int. Conf. Neural Networks Signal Process.*, vol. 2, pp. 1062–1065, Dec. 2003.
- [94] S. Kondo, "Reduction of blocking artifacts in MPEG-2 video using a block classification technique," in *Proc. Int. Conf. Consumer Electron. Digest of Tech. Papers*, pp. 238–239, Jun. 2000.
- [95] G. A. Triantafyllidis, D. Tzovaras, D. Sampson, and M. G. Strintzis, "A hybrid algorithm for the removal of blocking artifacts," in *Proc. IEEE Int. Conf. Multimedia and Expo*, vol. 1, pp. 161–164, Aug. 2002.
- [96] Y. L. Lee, H. C. Kim, and H. W. Park, "Blocking effect reduction of JPEG images by signal adaptive filtering," *IEEE Trans. Image Process.*, vol. 7, no. 2, pp. 229–234, Feb. 1998.
- [97] T. S. Liu and N. Jayant, "Adaptive post-processing algorithms for low rate video signals," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, vol. 4, no. 7, pp. 1032–1035, 1995.
- [98] J. X. Yang and W. H. Ren, "A non-linear post filtering method for flicker reduction in H.264/AVC coded video sequences," *IEEE 10th Workshop Multimedia Signal Process.*, pp. 181–186, Oct. 2008.

- [99] D. T. Vo and T. Q. Nguyen, "Quality enhancement for motion JPEG using temporal redundancies," in *Proc. IEEE Int. Conf. Image Process.*, vol. 4, pp. 389–392, Sept. 2007.
- [100] I. R. Ismaeil and R. K. Ward, "Removal of DCT blocking artifacts using DC and AC filtering," in *Proc. IEEE Pacific Rim Conf. Commun. Computers Signal Process.*, vol. 1, pp. 229–232, Aug. 2003.
- [101] B. Macq, M. Mattavelli, O. V. Claster, E. V. D. Plancke, S. Comes, and W. Li, "Image visual quality restoration by cancellation of the unmasked noise," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, vol. 5, pp. 53–56, Apr. 1994.
- [102] Y. Nakajima, H. Hori, and T. Kanoh, "A pel adaptive reduction of coding artifacts for MPEG video signals," in *Proc. IEEE Int. Conf. Image Process.*, vol. 2, pp. 928–932, Nov. 1994.
- [103] S. W. Hong, Y. H. Chan, and W. C. Siu, "A practical real-time post-processing technique for block effect elimination," in *Proc. IEEE Int. Conf. Image Process.*, vol. 1, pp. 21–24, Sept. 1996.
- [104] O. G. Guleryuz, "Linear, worst-case estimators for denoising quantization noise in transform coded images," *IEEE Trans. Image Process.*, vol. 15, no. 10, pp. 2967–2986, Oct. 2006.
- [105] S. Yang and T. Q. Nguyen, "Denoising in the lapped transform domain," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, vol. 6, pp. 173–176, Apr. 2003.
- [106] J. Luo, C. W. Chen, K. J. Parker, and T. Huang, "Artifact reduction in low bit rate DCT-based image compression," *IEEE Trans. Image Process.*, vol. 5, no. 9, pp. 1363–1368, 1996.
- [107] J. Mateos, A. Katsaggelos, and R. Molina, "A Bayesian approach for estimation and transmission of regularization parameters for reducing blocking artifacts," *IEEE Trans. Image Process.*, vol. 9, no. 7, pp. 1200–1215, Jun. 2000.
- [108] Z. Li and E. J. Delp, "MAP-based post processing of video sequences using 3-D Huber-Markov random field model," in *Proc. IEEE Int. Conf. Multimedia and Expo*, vol. 1, pp. 153–156, Aug. 2002.
- [109] J. Li and C. C. J. Kuo, "Coding artifact removal with multiscale post-processing," *IEEE Trans. Image Process.*, vol. 1, pp. 45–48, Oct. 1997.

- [110] L. Yehong and X. Lin, "Blind image restoration with eigen-face subspace," *IEEE Trans. Image Process.*, vol. 14, no. 11, pp. 1766–1772, Nov. 2005.
- [111] S. W. Hong, Y. H. Chan, and W. C. Siu, "Subband adaptive regularization method for removing blocking effect," *IEEE Trans. Image Process.*, vol. 2, pp. 523–526, Oct. 1995.
- [112] J. J. Zou and H. Yan, "A deblocking method for BDCT compressed images based on adaptive projections," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 3, pp. 430–435, Mar. 2005.
- [113] H. Paek and S. U. Lee, "A projection-based post-processing technique to reduce blocking artifact using a priori information on DCT coefficients of adjacent blocks," *IEEE Trans. Image Process.*, vol. 2, pp. 53–56, Sept. 1996.
- [114] T. Kartalov, Z. A. Ivanovski, L. Panovski, and L. J. Karam, "An adaptive POCS algorithm for compression artifacts removal," in *Proc. Int. Symp. Signal Process. Applications*, pp. 1–4, Feb. 2007.
- [115] M. G. Choi, Y. Yang, and N. P. Glatsanos, "Multichannel regularized recovery of compressed video sequences," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 48, no. 4, pp. 376–387, Dec. 2001.
- [116] D. C. Youla, "Post-processing technique based on POCS for visual enhancement in HDTV images," *IEEE Trans. Consumer Electron.*, vol. 47, no. 3, pp. 652–659, Aug. 2001.
- [117] L. Zhen and E. J. Delp, "Blocking artifact reduction using a transform-domain Markov random field model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 12, pp. 1583–1593, Dec. 2005.
- [118] A. W. C. Liew, Y. Hong, and N. F. Law, "POCS-based blocking artifacts suppression using a smoothness constraint set with explicit region modeling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 6, pp. 795–800, Jun. 2005.
- [119] C. Weerasinghe, W. C. Liew, and H. H. Yan, "Artifact reduction in compressed images based on region homogeneity constraints using the projection onto convex sets algorithm," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 10, pp. 891–897, Oct. 2002.

- [120] K. Y. Kwak and R. Haddad, "Projection-based eigenvector decomposition for reduction of blocking artifacts of DCT coded image," *IEEE Trans. Image Process.*, vol. 2, pp. 527–530, Oct. 1995.
- [121] Y. K. Lai, J. Li, and C. C. J. Kuo, "Image enhancement for low bit rate JPEG and MPEG coding via postprocessing," in *SPIE: Visual Commun. Image Process.*, Mar. 1996.
- [122] A. N. Netravali and B. G. Haskell, *Digital pictures: Representation and compression*. New York, London: Plenum Press, 2nd ed., 1995.
- [123] Y. Wang, Y. Wang, and H. Kuroda, "A globally adaptive pixel-decimation algorithm for block-motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 6, pp. 1006–1011, Sept. 2000.
- [124] J. Kim and T. Choi, "A fast full-search motion-estimation algorithm using representative pixels and adaptive matching scan," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 7, pp. 1040–1048, Oct. 2000.
- [125] A. Erturk and S. Erturk, "Two-bit transform for binary block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 7, pp. 938–946, Jul. 2005.
- [126] B. Demir and S. Erturk, "Block motion estimation using adaptive modified two-bit transform," *IET Image Process.*, vol. 1, no. 2, pp. 215–222, Jun. 2007.
- [127] C. B. Wu, C. Y. Yao, B. D. Liu, and J. F. Yang, "DCT-based adaptive thresholding algorithm for binary motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 5, pp. 694–703, May 2005.
- [128] I. H. Lee and R. H. Park, "A fast block matching algorithm using integral projection," in *Proc. Tencon. IEEE Region 10 Conf.*, pp. 590–594, Aug 1987.
- [129] J. S. Kim and R. H. Park, "Feature-based block matching algorithm using integral projections," *IEEE J. Select. Areas Commun.*, vol. 10, no. 5, pp. 968–971, Jun. 1992.
- [130] Y. C. Lin and S. C. Tai, "Fast full-search block-matching algorithm for motion-compensated video compression," *IEEE Trans. Commun.*, vol. 45, no. 5, pp. 527–531, May 1997.

- [131] V. L. Do and K. Y. Yun, "A low-power VLSI architecture for full-search block-matching motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 4, pp. 393–398, Aug. 1998.
- [132] P. Viola and M. J. Jones, "Robust real-time object detection," *Technical Report CRL 2001/01*, Cambridge Research Laboratory, Feb. 2001.
- [133] X. Wu, "Image coding by adaptive tree-structured segmentation," *IEEE Trans. Info. Theory*, vol. 38, no. 6, pp. 1755–1767, Nov. 1992.
- [134] X. Wu, "Adaptive split-and-merge segmentation based on piecewise least-square approximation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, no. 8, pp. 808–815, Aug. 1993.
- [135] X. Wu, "Efficient statistical computations for optimal color quantization," *Graphic Gems, vol. II. Academic Press, New York*, pp. 126–133, 1991.
- [136] MPEG Software Simulation Group (MSSG), "Test Model 5 (TM5)," [Online] Available: <http://www.mpeg.org/MPEG/MSSG/>.
- [137] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, and T. Wedi, "Video coding with H.264/AVC: tools, performance, and complexity," *IEEE Circuits and Systems Magazine*, vol. 4, no. 1, pp. 7–28, Apr. 2004.
- [138] S. J. Yu, J. S. Myung, D. G. Sim, and S. J. Oh, "Enhanced transform domain intra prediction for MPEG-2 to H.264/AVC transcoding," *IEEE Trans. Consumer Electron.*, vol. 54, no. 2, pp. 703–708, May 2008.
- [139] H. Kalva and B. Petljanski, "Exploiting the directional features in MPEG-2 for H.264 intra transcoding," *IEEE Trans. Consumer Electron.*, vol. 52, no. 2, pp. 706–711, May 2006.
- [140] C. H. Fu, Y. L. Chan, and W. C. Siu, "Fast motion estimation and mode decision for H.264 reverse transcoding," *Electron. Letters*, vol. 42, no. 24, pp. 1385–1386, Nov. 2006.
- [141] Z. Zhou, S. Sun, S. Lei, and M. T. Sun, "Coding mode decision for high quality MPEG-2 to H.264 transcoding," in *Proc. IEEE Int. Conf. Image Process.*, vol. 4, pp. 77–80, Oct. 2007.
- [142] J. Xin, J. Li, A. Vetro, H. Sun, and S. Sekiguchi, "Motion mapping for MPEG-2 to H.264/AVC transcoding," in *Proc. IEEE Int. Symp. Circuits and Systems*, pp. 1991–1994, May 2007.

- [143] H. Kalva and P. Kunzelmann, "Dynamic motion estimation for transcoding P frames in H.264 to MPEG-2 transcoders," *IEEE Trans. Consumer Electron.*, vol. 54, no. 2, pp. 657–662, May 2008.
- [144] G. F. Escribano, H. Kalva, P. Cuenca, L. O. Barbosa, and A. Garrido, "A fast MB mode decision algorithm for MPEG-2 to H.264 P-frame transcoding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 2, pp. 172–185, Feb. 2008.
- [145] H. Zhihua and M. Bystrom, "Improved conversion from DCT blocks to integer cosine transform blocks in H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 6, pp. 851–857, Jun. 2008.
- [146] P. Maria, N. Ling, and W. Shang, "Coefficient conversion for transform domain VC-1 to H.264 transcoding," *IEEE Workshop on Signal Process. Systems*, pp. 363–367, Oct. 2007.
- [147] Z. G. Li, F. Pan, K. P. Lim, G. N. Feng, X. Lin, and S. Rahardja, "Adaptive basic unit layer rate control for JVT," JVT-G012-r1, 7th Meeting, Pattaya II, Thailand, March 2003.
- [148] A. Joch, F. Kossentini, and P. Nasiopoulos, "A performance analysis of the ITU-T draft H.26L video coding standard," *12th International Packetvideo Workshop*, Apr. 2002.
- [149] F. Pan, X. Lin, and R. Susanto, "Fast intra mode decision algorithm for H.264/AVC video coding," in *Proc. IEEE Int. Conf. Image Process.*, vol. 2, pp. 781–784, Oct. 2004.
- [150] B. Shen and I. K. Shethi, "Direct feature extraction from compressed images," *Storage and Retrieval for Image and Video Databases*, 1996.
- [151] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, pp. 560–576, Jul. 2003.
- [152] B. Jeon and J. Lee, "Fast mode decision for H.264," *ITU-T VCEG (Q.6/16)*, Jul. 2003.
- [153] B. Xie and W. Zeng, "Sequence-based rate control for constant quality video," in *Proc. IEEE Int. Conf. Image Process.*, vol. 1, pp. 77–80, 2002.
- [154] ITU-T/SG15, "Video codec test model TMN-8," Portland, June 1997.

- [155] JM Reference Software Version 7.4 <http://bbs.hhi.de/suehring/tml/download>.
- [156] R. Hardie, "A fast image super-resolution algorithm using an adaptive Wiener filter," *IEEE Trans. Image Process.*, vol. 16, no. 12, pp. 2953–2964, Dec. 2007.
- [157] G. H. Costa and J. C. M. Bermudez, "Statistical analysis of the LMS algorithm applied to super-resolution image reconstruction," *IEEE Trans. Signal Process.*, vol. 55, no. 5, pp. 2084–2095, May 2007.
- [158] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau, "Multiframe resolution enhancement methods for compressed video," *IEEE Signal Process. Letters*, vol. 9, no. 6, pp. 170–174, Jun. 2002.
- [159] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau, "Multiframe blocking-artifact reduction for transform-coded video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 4, pp. 276–282, Apr. 2002.
- [160] H. Shen, L. Zhang, B. Huang, and P. Li, "A MAP approach for joint motion estimation, segmentation, and super resolution," *IEEE Trans. Image Process.*, vol. 16, no. 2, pp. 479–490, Feb. 2007.
- [161] C. A. Segall, R. Molina, and A. K. Katsaggelos, "High-resolution images from low-resolution compressed video," *IEEE Signal Process. Magazine*, vol. 20, no. 3, pp. 37–48, May 2003.
- [162] I. Richardson, *H.264 and MPEG-4 video compression*. John Wiley & Sons, 2003.
- [163] R. L. de Queiroz, "Processing JPEG-compressed images and documents," *IEEE Trans. Image Process.*, vol. 7, no. 12, pp. 1661–1672, 1998.
- [164] F. Muller, "Distribution shape of two-dimensional DCT coefficients of natural images," *IEEE Electron. Letters*, vol. 29, no. 22, pp. 1935–1936, 1993.
- [165] T. Eude, R. Grisel, H. Cherifi, and R. Debrie, "On the distribution of the DCT coefficients," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, vol. 5, pp. 365–368, Oct. 1994.
- [166] E. Y. Lam and I. W. Goodman, "A mathematical analysis of the DCT coefficient distribution for images," *IEEE Trans. Image Process.*, vol. 9, no. 10, pp. 1661–1666, Oct. 2000.

- [167] S. R. Smooth and R. A. Lowe, "Study of DCT coefficient distribution," in *Proc. SPIE*, pp. 403–408, 1996.
- [168] N. Kamaci, Y. Altunbasak, and R. M. Mersereau, "Frame bit allocation for the H.264/AVC video coder via Cauchy-density-based rate and distortion model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 8, pp. 994–1006, Aug. 2005.
- [169] T. Brandao and M. P. Queluz, "No-reference image quality assessment based on DCT domain statistics," *Signal Processing: Image Commun.*, vol. 88, no. 4, pp. 822–833, 2008.
- [170] Y. P. Tan, S. R. Kulkarni, and P. J. Ramadge, "A framework for measuring video similarity and its application to video query by example," in *Proc. IEEE Int. Conf. Image Process.*, vol. 2, pp. 106–110, Oct. 1999.