Fast algorithms for H.264 encoding and transcoding

Ye, Tianxiao

2009

Ye, T. (2009). Fast algorithms for H.264 encoding and transcoding. Master' s thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/15162>

<https://doi.org/10.32657/10356/15162>

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

FAST ALGORITHMS FOR H.264
ENCODING AND TRANSCODING

**FAST ALGORITHMS FOR H.264
ENCODING AND TRANSCODING**

YE TIANXIAO

YE TIANXIAO

SCHOOL OF ELECTRICAL & ELECTRONIC ENGINEERING

2009

2009

FAST ALGORITHMS FOR H.264 ENCODING AND TRANSCODING

YE TIANXIAO

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement for the degree of
Master of Engineering

2009

Acknowledgements

I would like to first thank my supervisor Associate Professor Xue Ping, for his kind guidance, continuous support, and valuable suggestions in my graduate study and research work.

I would want to express my sincere gratitude to Associate Professor Tan Yap-Peng for his invaluable advice, fruitful discussion and encouragement. I learned a lot from his methodology, enthusiasm and earnest attitude in doing research.

I am grateful for all members of Media Technology Lab for their support and assistant during my studies. I am very thankful to Dr. Chen Zhenzhong for his help in my paper writing.

Finally, I'd like to thank my father Ye Ximing, mother Gao Meiqin, and my wife Yang Weining for their love, support, and help over these years. I owe my deepest thankfulness to them for their dedication to my education.

Table of Contents

List of Figures.....	IV
List of Tables	V
Summary.....	VI
1 Introduction.....	1
1.1 Motivation.....	1
1.2 Objectives	3
1.3 Contributions.....	4
2 Overview of H.264.....	5
2.1 The enhanced and new coding features in H.264	5
2.2 Previous approaches for encoder complexity reduction	9
2.3 DCT, INT, HT, and Quantization in H.264	10
3 AZB Detection Algorithm for H.264 INT Using SAD Information	13
3.1 Motivation.....	13
3.2 Review of related work.....	15
3.2.1 Sousa's algorithm.....	16
3.2.2 Moon's algorithm.....	17
3.2.3 Su's algorithm and Wang's algorithm	18
3.2.4 Summary of previous methods	19
3.3 Maximum magnitude probability analysis.....	19
3.4 Efficient AZB detection algorithm	24
3.5 Efficient multi-strategy AZB detection algorithm.....	25
3.6 Comparison of AZB detection algorithms.....	28
3.7 Computational complexity.....	30

3.8	Experimental results.....	32
3.9	Summary	41
4	AZB Detection Algorithm for H.264 INT Using HT Information.....	42
4.1	Motivation.....	42
4.2	Review of related work.....	43
4.3	Correlation analysis for INT and HT coefficients	43
4.4	Efficient AZB detection algorithm by using HT information	46
4.5	Computational complexity.....	51
4.6	Experimental results.....	52
4.7	Summary	57
5	Low Complexity H.263 to H.264 Inter Frame Transcoding	58
5.1	Motivation.....	58
5.2	Design issue: transcoding architecture selection	59
5.3	Design issue: H.264 feature selection.....	61
5.4	Review on motion estimation and inter mode decision procedure	65
5.5	Proposed fast algorithms for low complexity transcoding	67
5.5.1	Fast skip mode estimation.....	68
5.5.2	Joint motion refinement & inter mode decision	72
5.6	Experimental results.....	75
5.7	Summary	79
6	Conclusion and Future Recommendation	80
6.1	Conclusion	80
6.2	Future Directions	81
	Bibliography	84
	Publication List	93

List of Figures

Figure 3.1: Coefficient grouping in a 4×4 transformed block.	23
Figure 3.2: Flowchart of the proposed EAZB+GM.	27
Figure 3.3: Fast implementation of INT and IINT.	30
Figure 3.4: Complexity required by the encoding schemes with AZB detection algorithm, compared with the original H.264 encoding scheme.	37
Figure 3.5: Proportion of the complexity of the AZB detection algorithm and the INT, Q, IQ, IINT operations, when HM [40], EAZB and EAZB+GM are implemented for the <i>Susie</i> sequence.	40
Figure 4.1: Flowchart of the proposed HTAZBD.	50
Figure 4.2: Complexity required by the encoding schemes with SATDAZBD and HTAZBD, compared with the original H.264/AVC encoding scheme.	54
Figure 5.1: An example of H.263 to H.264 cascaded pixel-domain transcoder.	60
Figure 5.2: An example of H.263 to H.264 transform-domain transcoder.	60
Figure 5.3: Flowchart of the proposed H.263 to H.264 inter frame transcoding scheme with the proposed fast algorithms.	68
Figure 5.4: Proportion of skip mode MB for <i>Susie</i> , <i>Foreman</i> , and <i>Mobile</i>	69
Figure 5.5: Flowchart of the proposed fast skip mode estimation.	71
Figure 5.6: The spatial-temporal candidate MVs in current frame and previous frame. ...	72
Figure 5.7: Sample frame from the <i>Flower</i> sequence.	78

List of Tables

Table 3.1: Theoretical 4×4 MMP_{DCT} matrixes for $\rho = 0.40$, $\rho = 0.60$, and $\rho = 0.75$.	22
Table 3.2: MMP'_{DCT} of two test sequences: <i>Foreman</i> and <i>Mobile</i> .	22
Table 3.3: $GMMP_{DCT}(i)$ for the typical range of correlation coefficient ρ .	24
Table 3.4: Computational complexity of INT, Q, IQ, IINT for different INT strategies.	30
Table 3.5: Computational complexity and the number of INT strategies of AZB detection algorithms.	32
Table 3.6: Comparison of FRR and FAR for AZB detection algorithms.	33
Table 3.7: PSNR and rate degradation of the lossy AZB detection algorithms.	35
Table 4.1: Correlation coefficient between INT and HT coefficients for the typical range of pixel correlation coefficient ρ .	45
Table 4.2: Computational complexity and the number of INT strategies for the HT information based AZB detection algorithms.	51
Table 4.3: Comparison of FRR and FAR for the HT information based AZB detection algorithms.	52
Table 4.4: PSNR and rate degradation of the HT information based AZB detection algorithms.	53
Table 5.1: Comparison of the features in H.263 and H.264 Baseline Profile.	62
Table 5.2: H.264 features applied in the proposed transcoder.	64
Table 5.3: PSNR results and encoding time obtained by transcoding H.263 sequences using four transcoding schemes.	76
Table 5.4: Performance of the proposed scheme with/without fast skip mode estimation.	78
Table 5.5: Performance of the proposed scheme with/without quarter-pel refinement.	79

Summary

The rapid growth of multimedia technology and telecommunication technology enables us to enjoy video services anywhere and anytime. In order to ensure satisfactory video service experiences, a number of video coding standards are introduced to meet different requirements of end users. Recently, a state-of-the-art video coding standard, H.264/AVC, emerges as one of the hottest technologies in the multimedia industry, due to its excellent data compression performance. The superior performance of this new standard also raises a need for transcoding videos in other formats to this new format in order to save the storage cost and the bandwidth cost. However, H.264/AVC is estimated three times more complex than the previous standards for general implementation and this high complexity issue becomes the main obstacle for its industry adoption. Therefore, it is essential to reduce the complexity of H.264 encoding and transcoding.

This thesis investigates several technologies to speed up H.264 encoder and transcoder, while maintaining a good video quality.

In the first part of this study, an all-zero block detection algorithm for H.264 is proposed by using the sum of absolute (SAD) information. This proposal is based on our observation that a large amount of all-zero blocks exist during encoding, which is due to the high compression property of H.264. An efficient method is proposed to detect these all-zero blocks in an early stage. With the successful early detection, the complexity of coding these blocks can be saved without video quality loss.

In the second part of this study, the relationship between integer transform and Hadamard transform is investigated and an effective scheme is presented to reuse Hadamard transform information for all-zero block detection. This scheme can be used when the SAD information is not available for previous SAD based all-zero block detection algorithms.

In the third part of this study, a low complexity H.263 to H.264 inter frame transcoding scheme is proposed. The early stop strategy is implemented in the skip mode detection. The spatial-temporal motion information is used in motion vector (MV) refinement and a simplified procedure of inter mode decision is proposed. The proposed fast algorithms largely speed up the process of inter frame transcoding with minor quality loss.

Chapter 1

Introduction

1.1 Motivation

In the last two decades, there have been significant advancements in video coding technique, which fertilized the bloom of such multimedia industries as the film industry, the telecommunications industry, and the streaming industry. Today, we can buy a Blu-Ray [1] disk (a 50 GB disc that uses a 405nm blue laser) player and watch the high-definition movie in our living rooms; we can use skype [2] (a software program that allows users to make telephone calls over the Internet) to video-chat with our friends on our laptops; we can record our favorite video clips and upload them to Youtube [3] (a famous video streaming website) to share with other people in the world. The enjoyment and distribution of media is tightly coupled with our daily life.

The driving force of these multimedia applications is a common video standard, which reduces the costs associated with redundant development and guarantees the unrestricted flow of user's content. In order to meet the needs of our exploding information society, the video coding standards are evolving quickly to achieve a better tradeoff between the quality (the image quality, the streaming stability, the user-oriented scalability, etc) and

Chapter 1 Introduction

the cost (the storage cost, the bandwidth cost, the power consumption cost, etc). H.261 [4] was developed in 1993 for video transmission over Integrated Service Digital Network (ISDN) lines on where data rates are multiples of 64 Kbit/s. MPEG-1[5] was published in 1993 to be optimized for encoding entertainment video at about 1.5 Mb/s, which is generally implemented in VCD (Video CD). MPEG-2 [6] was designed for higher quality applications which can afford higher bit-rates. The standard was published in 1995 and successfully created the DVD (digital video disc) industry, which revolutionized the way people watch video at home. H.263 [7] is a high compressed format, which focuses on video streaming and video conferencing over the Internet. Its first version was completed in 1995 and was further enhanced in projects known as H.263+ [7] and H.263++ [7]. MPEG-4 part 2 [8] is a content based format, which can support video coding for a wider range of bit-rates. The concepts of Video Object (VO) and Video Object Plane (VOP) have been introduced in MPEG-4 so that video conferencing at very low bit-rate becomes possible.

Generally, the compression performance of a new video standard is improved along with an increase of computational cost. H.264/AVC [9], as the latest video coding standard, significantly outperforms previous standards in terms of coding efficiency. It has been reported [10] that, in most cases, H.264 improves the coding efficiency by more than 2 dB when compared to the other standards. However, the performance gain of H.264 is achieved at the expense of tremendously increased coding complexity. Thus, there is a

Chapter 1 Introduction

strong need in complexity reduction to facilitate and accelerate the industry adoption of this new video coding standard.

The stunning compression performance of H.264 also stimulates the demand for transcoding the conventional format videos to this new one, so that the storage cost and the streaming cost can be reduced. An efficient transcoding scheme can extract the available information from the pre-coded stream and wisely use it to generate a good quality stream with acceptable computational cost. However, the new features and the high complexity of H.264 become the barrier of such efficient transcoding.

1.2 Objectives

The purpose of this research is to improve the efficiency of the H.264 encoding and transcoding so that the computational complexity can be reduced with negligible visual quality loss.

The main objective of this thesis is to investigate the unique features of H.264, especially for the integer transform [11], the motion refinement and the mode decision parts, in order to making good use of the uniqueness of H.264 in improving the encoding and transcoding efficiency.

Chapter 1 Introduction

1.3 Contributions

The contributions of this thesis are summarized as follows:

- The integer transform and quantization process of H.264 are investigated. Previous related AZB detection approaches are reviewed and compared. A mathematical model, called maximum magnitude probability model, is introduced to propose an efficient all-zero block detection algorithm by using SAD information.
- The relationship between integer transform and Hadamard transform is studied and an all-zero block detection algorithm is proposed by reusing Hadamard transform information, in case when SAD is not available.
- The choice of transcoding architecture and the H.264 features is analyzed. Then, a low-complexity H.263 to H.264 inter frame transcoder is proposed, which employs a fast skip mode estimation algorithm and a joint motion refinement & inter mode decision scheme.

Chapter 2

Overview of H.264

2.1 The enhanced and new coding features in H.264

H.264/AVC is the newest video coding standard developed by JVT (Joint Video Team) [12] consisting of experts from the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG). The standard was initiated in 1997 and the first version of the standard was finalized in 2003. Seven Profiles (Baseline, Main, Extended, etc) and several Levels, which set the limits for data bit-rate, frame size, picture buffer size, are defined in the standard in order to cover the wide range of coding application areas including high-definition TV broadcast, automatic surveillance system and real-time video conferencing. Recently, the eighth version of the standard was available, which contains the amendment for Scalable Video Coding (SVC) [13].

H.264 is widely regarded as an inflection point for the industry because it shows significant compression performance gain over contemporary video coding standards [14], while delivering a substantial improvement in perceived video quality. The basic functional elements of H.264 are the same as previous standards (MPEG-1, MPEG-2, MPEG-4 part 2, H.261 and H.263), i.e., transform coding for reduction of spatial

Chapter 2 Overview of H. 264

redundancy, scalar quantization with rounding operation for bit-rate control, inter-picture prediction for reduction of temporal redundancy, and entropy encoding for reduction of statistical redundancy. However, the enhancement of the detail of these traditional functional elements and the introduction of some new features are the true power of its superb coding efficiency.

4×4 integer transform and quantization: In contrast to the traditional 8×8 discrete cosine transform (DCT), the 4×4 integer transform (an integer approximation of 4×4 DCT) eliminates the dependency on a floating point implementation, such that the drift problem between encoder and decoder is eliminated. The scaling factors between original 4×4 DCT transform and this approximated version are absorbed in the quantization process, which guarantees the low complexity of this new transform with only shift and addition operation. Furthermore, the smaller block-size transform reduces the chances of “ringing” artifacts and “block” artifacts which are parlous for video perceptual quality.

Intra prediction: In compared with the direct residual coding for intra macroblock (MB) in many previous standards or the very simple transform domain intra prediction in MPEG-2 and MPEG-4, H.264 employs nine directional 4×4 intra prediction modes and four directional 16×16 intra prediction modes to form predictions of pixel values as linear interpolations of pixels from the adjacent decoded MBs. In such way, the spatial redundancy of intra MB is effectively reduced.

Chapter 2 Overview of H. 264

Multiple inter mode with variable block size: The selection of motion compensation block sizes in H.264 is more flexible than any previous standard. It ranges from 4×4 , as small as the basic block unit, to 16×16 , as large as the whole MB. There are nine inter modes defined in the standard with respect to the combination of various block sizes in a single inter MB. The choice of partition size, in another word, the choice of inter mode, is a tradeoff between the number of bits (the overhead bits for coding MV, the overhead bits for coding reference frame selection, the residual bits, etc.), and the distortion (the sum of squared difference for both luma and chroma part)

Quarter-pel motion compensation: H.264 elevates the MV accuracy onto the quarter-pel level, compared with the half-pel level of most prior standards. The half-pel samples are generated by using 6-tap FIR filter, and the quarter-pel samples are produced using bilinear interpolation. Quarter-pel accuracy outperforms half-pel accuracy especially at high bitrates and high video resolutions.

Multiple reference pictures: Instead of only one previous picture for predicting the pixel values in an incoming picture, H.264 uses multiple reference pictures (MCP) with the enhanced reference picture selection scheme to increase its prediction accuracy. MCP has been observed [15] to be very useful for decreasing the temporal redundancy caused by camera shaking, lighting changes, and sensor noise. Since MCP can result in a high storage requirement for encoding/decoding, the number of reference frames is strictly constrained by Profile or Level predefined in H.264 standard.

Chapter 2 Overview of H. 264

Rate-distortion optimized motion estimation, reference frame selection, and mode

decision: With so many options being involved in coding (e.g., the motion estimation in quarter-pel accuracy, the reference frame selection in MCP, the mode decision for choosing only one mode from the whole thirteen intra modes plus nine inter modes), the rate-distortion optimization (RDO) technique becomes the major practice for synthesizing the above powerful features and unleashing the full potential of H.264. However, the computational cost of RDO makes it a common bottleneck for H.264 implementation. Although an exhaustive RDO method is recommended in the reference software [16] to obtain the best rate-distortion performance, a lot of methods have been published [17]–[20] improving the efficiency of RDO.

Entropy coding: The context-adaptive variable-length coding (CAVLC) and the context-adaptive binary arithmetic coding (CABAC) are adopted in H.264 for entropy coding. CABAC is reported [21] to achieve about 9–14% compression gain over CAVLC with increased implementation complexity.

Deblocking filter: H.264 specifies the use of an adaptive deblocking filter that operates within the prediction loop in order to remove the artifacts caused by block prediction errors.

2.2 Previous approaches for encoder complexity reduction

It can be observed the compression gain and complexity pain come in pairs for almost all H.264 features. To keep the gain while ease the pain, optimizations are used at all of the implementation phases, including compiler and code optimization [22], taking advantage of certain architectural features of the machine [23], and algorithmic enhancements. Our work is focused on algorithm enhancements for H.264. Generally, motion estimation (ME) and mode decision are believed to be the most time consuming modules, many fast motion estimation (ME) algorithms and fast mode selection algorithms have been published.

For fast ME algorithms, a lot of search strategies were proposed to reduce the search points, including three-step search [24], new three-step search [25], cross search [26], diamond search [27], successive elimination algorithm [28], partial distortion elimination [29], and global elimination [30]. Furthermore, the main concepts of these conventional methods were implemented into H.264 with promising results [31]–[33].

As for fast mode decision algorithms, the general methodology being used is to cut out the complexity of checking unnecessary inter modes or intra modes, basing on the features extracted from the video stream. For instance, Yin [34] proposed a scheme to rearrange the mode test order such that the rate-distortion (RD) cost of the tested mode is used to decide the candidate modes to be tested further. In [35], Wu proposed to use

Chapter 2 Overview of H. 264

Sobel operator to estimate the spatial homogeneity and use pixel-domain MB difference to estimate temporal stationary characteristics for each MB so that only a limited number of inter prediction modes are tested for RDO.

As the ME and mode selection are optimized, the integer transform and quantization, which are the important process of RDO, should be the target of complexity reduction. In our research on H.264 encoding, we propose some fast algorithms for these two processes to avoid major portion of unnecessary computation.

2.3 DCT, INT, HT, and Quantization in H.264

In this section, we give some introductions and notations for H.264 residual coding procedure. This background information will be useful for our presentation in the following chapters.

In H.264, the 4×4 INT is used as an integer approximation of the 4×4 DCT, allowing for bit-exact implementation for all encoders and decoders, thus solving the drift problem of DCT. The 4×4 Hadamard transform (HT) is used as an option during the sub-pixel motion estimation to achieve a better estimation of MVs.

Three types of two-dimensional transforms of a 4×4 residual block $a(x, y)$ can be uniformly defined as

Chapter 2 Overview of H. 264

$$\tilde{a}_{type}(u, v) = \sum_{x=0}^3 \sum_{y=0}^3 a(x, y) \cdot \phi_{type}(x, u) \cdot \phi_{type}(y, v) \quad , \quad for \quad 0 \leq u \leq 3, 0 \leq v \leq 3 \quad (2.1)$$

with

$$\phi_{DCT}(m, n) = c_m \cdot \cos\left(\frac{2m+1}{8}n\pi\right), \quad \phi_{INT} = \langle 2.5 \cdot \phi_{DCT} \rangle = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & 1 \end{bmatrix}, \quad \phi_{HAT} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & 1 \end{bmatrix} \quad (2.2)$$

where $c_0 = 1/2$ and $c_m = 1/\sqrt{2}$ for $m > 0$; $\langle \rangle$ denotes the rounding-off operation;

ϕ_{DCT} , ϕ_{INT} and ϕ_{HT} stand for the kernels of DCT, INT and HT, respectively.

With the quantization parameter denoted as Q_p , the quantized INT coefficient can be written as

$$\hat{a}_{INT}(u, v) = \text{sign}(\tilde{a}_{INT}(u, v)) \times (|\tilde{a}_{INT}(u, v)| \cdot M(Q_p \% 6, r) + f) >> qbits \quad (2.3)$$

where $>>$ indicates the binary shift right, $\%$ is the modular operator, $\lfloor \rfloor$ denotes the floor operation, $r = (u \% 2) + (v \% 2)$, $qbits = 15 + \lfloor Q_p / 6 \rfloor$, and

$$f = \begin{cases} 2^{qbits}/3 & \text{for intra block} \\ 2^{qbits}/6 & \text{for inter block} \end{cases} \quad (2.4)$$

Instead of a shared quantization coefficient for an 8×8 DCT block in MPEG-2 and H.263, the quantization coefficient $M(Q_p \% 6, r)$ is different for the frequency components in H.264/AVC.

Chapter 2 Overview of H. 264

$$M(Q_p \% 6, r) = \begin{bmatrix} 13107 & 8066 & 5243 \\ 11916 & 7490 & 4660 \\ 10082 & 6554 & 4194 \\ 9362 & 5825 & 3647 \\ 8192 & 5243 & 3355 \\ 7282 & 4559 & 2893 \end{bmatrix} \quad (2.5)$$

As we know, the 4×4 INT is an integer approximation of a 4×4 DCT. By using the different scaling factors absorbed in $M(Q_p \% 6, r)$, the INT coefficients will be scaled to be very close to its corresponding DCT coefficients during the quantization process. This scaling process is part of equation (2.3) and we define $\tilde{a}'_{DCT}(u, v)$ as the nearly-DCT coefficients, whose difference with the DCT coefficient $\tilde{a}_{DCT}(u, v)$ is negligible, i.e.

$$\tilde{a}'_{DCT}(u, v) = \tilde{a}_{INT}(u, v) \cdot M(Q_p \% 6, r) \cdot \tau_{Q_p} \approx \tilde{a}_{DCT}(u, v), \text{ for } 0 \leq u \leq 3, 0 \leq v \leq 3 \quad (2.6)$$

here τ_{Q_p} is a constant value depending on Q_p only.

Chapter 3

AZB Detection Algorithm for H.264 INT Using SAD Information

3.1 Motivation

In H.264 video stream, large amount of AZBs (all pixel values of the quantized INT block are equal to zero) are generated during encoding. Since no residual information is coded for AZB, the H.264 encoder can achieve a very high compression ratio with heavy distribution of AZBs, especially under tight bandwidth. Generally, AZBs can be divided as two types according to how they are produced. The first type is generated by skip mode, which is an ultra-low signaling cost mode defined in H.264 standard. The residual block is forced to be all-zero for skip mode without going through INT or quantization. The second type is caused by the quantization. When quantization step is large enough, all quantized INT coefficients become zero after the rounding operation, so as the residual block. The work in this chapter is to achieve the detection of the second kind of AZB. If such AZBs can be detected at an early stage, the computational complexity in coding of them can be saved without any RD performance penalty.

Chapter 3 AZB Detection Algorithm for H.264 INT Using SAD Information

Several AZB detection algorithms have been published by reusing the method of sum of absolute difference (SAD), which is the side information of the motion estimation. Sousa [36] proposed a sufficient condition for AZB detection in H.263 based on the SAD for luminance blocks and quantization step. If the SAD of the block satisfies the condition, the block is considered as AZB after transform and quantization. Therefore, only the motion vector should be recorded. The rest of coding steps such as DCT/IDCT, quantization and inverse quantization can be skipped and a lot of computation can be saved. Moon *et al.* [37] extended Sousa's idea to H.264/AVC. A residual block is divided into four regions. For each region, the sufficient condition was investigated based on the property of INT. This approach has been further extended to detect more AZBs in [38] and [39]. The basic idea of these algorithms is to check more INT coefficients, at the cost of increasing complexity, to obtain a more precise magnitude supreme for the corresponding nearly-DCT block (the INT coefficients will be scaled very close to DCT coefficients during the quantization process). More AZBs can thus be detected before performing INT. However, since the AZB detection is carried out before each INT, the complexity should be properly considered. Wang *et al.* [40] presented a Gaussian model (GM) based approach for early detection of AZBs. They further combined the lossless AZB detection algorithm with GM and derived a hybrid model algorithm (HM) [40]. GM and HM can detect more AZBs at the expense of potential PSNR loss.

Chapter 3 AZB Detection Algorithm for H.264 INT Using SAD Information

In this chapter we introduced a maximum magnitude probability (MMP) model to measure mathematically the contribution of individual transformed coefficients to AZB detection. An efficient AZB detection algorithm (EAZB) is proposed by dropping investigations on those transform coefficients with low contributions. We combine EAZB with GM [40] to obtain the EAZB+GM, which can further improve the multi-threshold AZB detection method.

The organization of this chapter is as follows: Section 3.2 reviews a few existing SAD based AZB detection algorithms. The MMP (maximum magnitude probability) model is introduced in Section 3.3. EAZB and EAZB+GM are proposed in Sections 3.4 and 3.5, respectively. The comparison of all these AZB detection algorithms is conducted in Section 3.6. The computational complexity issue is discussed in Section 3.7. Experiment results are given in Section 3.8 to show the advantages of EAZB and EAZB+GM. Section 3.9 draws our summary.

3.2 Review of related work

An INT coefficient will be quantized to zero, if its magnitude is smaller than the quantization step size [37]:

$$|\tilde{a}_{INT}(u, v)| < T_{INT}(r, Q_P), \text{ where } T_{INT}(r, Q_P) = \frac{2^{q_{bits}} - f}{M(Q_P \% 6, r)} \quad (3.1)$$

Chapter 3 AZB Detection Algorithm for H.264 INT Using SAD Information

In another word, an INT block will become AZB if the magnitude supreme of the corresponding nearly-DCT block is smaller than the threshold $T'_{DCT}(Q_p)$ as shown below

$$\text{MAX}_{0 \leq u \leq 3, 0 \leq v \leq 3} \left\{ \left| \tilde{a}'_{DCT}(u, v) \right| \right\} < T'_{DCT}(Q_p), \text{ where } T'_{DCT}(Q_p) = (2^{q_{bits}} - f) \cdot \tau_{Q_p} \quad (3.2)$$

Compared with $T_{INT}(r, Q_p)$, the dependence on frequency position r is eliminated in $T'_{DCT}(Q_p)$ so that the threshold checking in equation (3.2) becomes very simple. The methodology of AZB detection is to estimate the magnitude upper bound of each individual frequency component in order to predict the magnitude supreme of the nearly-DCT block and then compare it with a predefined threshold in the same way as (3.2) to decide whether it is an AZB.

3.2.1 Sousa's algorithm

In Sousa's algorithm [36], the magnitude upper bound of INT coefficients can be obtained by the inequality:

$$\begin{aligned} |\tilde{a}_{INT}(u, v)| &\leq \sum_{x=0}^3 \sum_{y=0}^3 |\phi_{INT}(x, u) \cdot \phi_{INT}(y, v)| \cdot |a(x, y)| \\ &\leq \text{MAX}_{0 \leq x \leq 3, 0 \leq y \leq 3} (|\phi_{INT}(x, u) \cdot \phi_{INT}(y, v)|) \cdot \sum_{x=0}^3 \sum_{y=0}^3 |a(x, y)| \end{aligned} \quad (3.3)$$

We define:

$$C_{Sousa}(r) = \text{MAX}_{0 \leq x \leq 3, 0 \leq y \leq 3} (|\phi_{INT}(x, u) \cdot \phi_{INT}(y, v)|), \quad SAD = \sum_{x=0}^3 \sum_{y=0}^3 |a(x, y)| \quad (3.4)$$

Chapter 3 AZB Detection Algorithm for H.264 INT Using SAD Information

With the INT kernel ϕ_{INT} defined in (2.2) and $M(Q_P \% 6, r)$ defined in (2.5), it can be easily derived that:

$$C_{Sousa}(0) = 1, C_{Sousa}(1) = 2, C_{Sousa}(2) = 4 \quad (3.5)$$

and

$$C_{Sousa}(0) \cdot M(Q_P \% 6, 0) < C_{Sousa}(1) \cdot M(Q_P \% 6, 1) < C_{Sousa}(2) \cdot M(Q_P \% 6, 2) \quad (3.6)$$

With (3.6), the magnitude supreme of the corresponding nearly-DCT block is obtained from the frequency components with $r = 2$ as

$$\begin{aligned} \max_{0 \leq u \leq 3, 0 \leq v \leq 3} \{|\tilde{a}'_{DCT}(u, v)|\} &= \max_{0 \leq u \leq 3, 0 \leq v \leq 3} \{|\tilde{a}_{INT}(u, v) \cdot M(Q_P \% 6, r)|\} \cdot \tau_{Q_P} \\ &= 4 \cdot M(Q_P \% 6, 2) \cdot SAD \cdot \tau_{Q_P} \end{aligned} \quad (3.7)$$

Considering (3.1), (3.2), and (3.7), the sufficient condition of Sousa's AZB detection method is derived as:

$$SAD < T_{Sousa}(Q_P), \text{ where } T_{Sousa}(Q_P) = \frac{1}{4} \cdot T_{INT}(2, Q_P) \quad (3.8)$$

3.2.2 Moon's algorithm

To improve Sousa's algorithm, Moon *et al.* [37] divided the residual block into four regions and calculated SADs of each region as $S_0, \dots, S_3$:

$$\begin{aligned} S_0 &= |a(0, 0)| + |a(0, 3)| + |a(3, 0)| + |a(3, 3)| \\ S_1 &= |a(0, 1)| + |a(0, 2)| + |a(3, 1)| + |a(3, 2)| \\ S_2 &= |a(1, 1)| + |a(1, 2)| + |a(2, 1)| + |a(2, 2)| \\ S_3 &= |a(1, 0)| + |a(1, 3)| + |a(2, 0)| + |a(2, 3)| \end{aligned} \quad (3.9)$$

Chapter 3 AZB Detection Algorithm for H.264 INT Using SAD Information

Considering the property of $I_{INT}(x, y, u, v)$, a much precise magnitude upper bound of INT coefficients in the frequency positions with $r = 2$ is obtained as:

$$|\tilde{a}_{INT}(u, v)| \leq 4 \cdot SAD - 2 \cdot \text{MIN}\{S_0 + S_3, S_1 + S_2\}, \text{ for } r = 2 \quad (3.10)$$

So the sufficient condition of Moon's AZB detection method becomes

$$SAD < T_{Moon}(Q_P) \quad (3.11)$$

$$\text{where } T_{Moon}(Q_P) = \text{MIN} \left\{ \frac{1}{2} \cdot T_{INT}(1, Q_P), \frac{1}{4} \cdot T_{INT}(2, Q_P) + \frac{1}{2} \cdot \text{MIN}\{S_0 + S_3, S_1 + S_2\} \right\}$$

3.2.3 Su's algorithm and Wang's algorithm

The ideas in [38] and [39] are similar to the method presented in [37]. More INT coefficients are checked to get a tighter magnitude upper bound so that more AZBs can be detected. However, higher computational complexity is introduced. The proposed thresholds $T_{Su}(Q_P)$ and $T_{Wang}(Q_P)$ are shown as follows:

$$T_{Su}(Q_P) = \text{MIN} \left\{ \begin{array}{l} T_{INT}(0, Q_P), \\ T_{INT}(1, Q_P) - 2 \cdot \text{MAX}_{0 \leq i \leq 3} \{S_i\}, \\ T_{INT}(2, Q_P) - 5 \cdot \text{MAX}_{0 \leq i \leq 3} \{S_i\} \end{array} \right\} \quad (3.12)$$

$$T_{Wang}(Q_P) = \text{MIN} \left\{ \begin{array}{l} T_{INT}(0, Q_P), \\ \frac{1}{2} \cdot T_{INT}(1, Q_P) + \frac{1}{2} \cdot \text{MIN}\{S_1 + S_2, S_0 + S_3, S_2 + S_3, S_0 + S_1\}, \\ \frac{1}{4} \cdot T_{INT}(2, Q_P) + \frac{1}{4} \cdot \text{MIN} \left\{ \begin{array}{l} 2S_1 + 3S_2 + 2S_3, 2S_0 + 2S_2 + 3S_3, \\ 2S_0 + 3S_1 + 2S_2, 3S_0 + 2S_1 + 2S_3 \end{array} \right\} \end{array} \right\} \quad (3.13)$$

Chapter 3 AZB Detection Algorithm for H.264 INT Using SAD Information

3.2.4 Summary of previous methods

In summary, $T_{Sousa}(Q_P)$ is based on the magnitude upper bound of the frequency components with $r=2$; $T_{Moon}(Q_P)$ is based on the magnitude upper bound of the frequency components with $r=1$ and $r=2$; $T_{Su}(Q_P)$ and $T_{Wang}(Q_P)$ are based on the magnitude upper bound of all frequency components.

3.3 Maximum magnitude probability analysis

We found that these algorithms [36]–[39] spend a large amount of computation to check the high frequency components, which are unlikely to hit the magnitude supreme of the corresponding nearly-DCT block. The problem becomes even worse when all frequency components are checked, as shown in equation (3.12) and (3.13). To address the above problem, we derive a mathematical model to study the contributions of different frequency components for the AZB detection.

We model a 4×4 residual block as a random field with variance σ^2 and a separable correlation coefficient ρ between the horizontally or vertically adjacent pixel values. The DCT coefficient, $\tilde{a}_{DCT}(u, v)$, is approximated by a zero mean Laplacian distributed variable [41] as

$$L\left(x, \sigma_{DCT}^2(u, v)\right) = \frac{1}{\sqrt{2} \cdot \sigma_{DCT}(u, v)} e^{\frac{-\sqrt{2}|x|}{\sigma_{DCT}(u, v)}} \quad (3.14)$$

Chapter 3 AZB Detection Algorithm for H.264 INT Using SAD Information

The variance of the (u, v) th DCT coefficient, $\sigma_{DCT}^2(u, v)$, can be written as [42]

$$\sigma_{DCT}^2(u, v) = \sigma^2 \cdot \lambda_{DCT}(u) \cdot \lambda_{DCT}(v) \quad (3.15)$$

where

$$\lambda_{DCT} = \text{Diag} \left(\phi_{DCT} \times R \times (\phi_{DCT})^T \right) \quad (3.16)$$

Here, $\text{Diag}(\)$ represents the diagonal of a matrix as a row vector, R is the correlation matrix of a column of the incoming residual block.

$$R = \begin{bmatrix} 1 & \rho & \rho^2 & \rho^3 \\ \rho & 1 & \rho & \rho^2 \\ \rho^2 & \rho & 1 & \rho \\ \rho^3 & \rho^2 & \rho & 1 \end{bmatrix} \quad (3.17)$$

$\sigma_{DCT}^2(u, v)$ can be further simplified as $\sigma_{DCT}^2(n)$, where $n = 4 \cdot v + u$. With Laplacian model, the probability, that the magnitude of the (u, v) th DCT coefficient is no larger than w , is equal to $\int_{-w}^w L(x, \sigma_{DCT}^2(n)) \cdot dx$. For the orthogonal transform like DCT, the variances of the transformed coefficients can be considered to be almost independent among all (u, v) . Thus, the probability that the (u, v) th DCT coefficient having the largest magnitude of the 4×4 block can be derived as,

$$MMP_{DCT}(u, v) = \int_{-\infty}^{\infty} L(x, \sigma_{DCT}^2(n)) \cdot \frac{\prod_{m=0}^{15} \left(\int_{-x}^x L(y, \sigma_{DCT}^2(m)) \cdot dy \right)}{\left[\int_{-x}^x L(y, \sigma_{DCT}^2(n)) \cdot dy \right]} \cdot dx \quad (3.18)$$

Chapter 3 AZB Detection Algorithm for H.264 INT Using SAD Information

where $MMP_{DCT}(u, v)$ is defined as the maximum magnitude probability (MMP) for the DCT coefficient at spatial frequency (u, v) .

As shown in equation (3.1) and (3.2), detecting AZB is equivalent to predicting the magnitude supreme of a 4×4 DCT block. A higher $MMP_{DCT}(u, v)$ indicates that the magnitude of the DCT coefficient at the spatial frequency (u, v) is more likely to be the magnitude supreme of the whole block. Thus, the coefficients at such spatial frequencies are more important for AZB detection. It should be noted that $MMP_{DCT}(u, v)$, instead of $MMP_{INT}(u, v)$, is used for the AZB detection in H.264, because the INT coefficients will be compensated to be nearly-DCT coefficients by $M(Q_p \% 6, r)$ during the quantization process.

For the DC coefficient, $MMP_{DCT}(0, 0)$ can be obtained as:

$$\begin{aligned}
 MMP_{DCT}(0, 0) &= 2 \cdot \int_0^\infty \left(\frac{1}{\sqrt{2}\sigma_{DCT}(0)} e^{\frac{-\sqrt{2}x}{\sigma_{DCT}(0)}} \right) \cdot \prod_{m=1}^{15} \left(1 - e^{\frac{-\sqrt{2}x}{\sigma_{DCT}(m)}} \right) \cdot dx \\
 &= \frac{\sqrt{2}}{\sigma_{DCT}(0)} \cdot \int_0^\infty e^{\frac{-\sqrt{2}x}{\sigma_{DCT}(0)}} \cdot \left(1 - \left(\sum_{m=1}^{15} e^{\frac{-\sqrt{2}x}{\sigma_{DCT}(m)}} \right) + \dots - \left(e^{\sum_{m=1}^{15} \frac{-\sqrt{2}x}{\sigma_{DCT}(m)}} \right) \right) \cdot dx \\
 &= \frac{1}{\sigma_{DCT}(0)} \cdot \left(\frac{1}{\frac{1}{\sigma_{DCT}(0)}} - \sum_{m=1}^{15} \left(\frac{1}{\frac{1}{\sigma_{DCT}(0)} + \frac{1}{\sigma_{DCT}(m)}} \right) + \dots - \left(\frac{1}{\frac{1}{\sigma_{DCT}(0)} + \sum_{m=1}^{15} \left(\frac{1}{\sigma_{DCT}(m)} \right)} \right) \right) \quad (3.19)
 \end{aligned}$$

Chapter 3 AZB Detection Algorithm for H.264 INT Using SAD Information

The MMP of the DCT coefficients in other frequency positions can be derived in a similar way as (3.19). MMP_{DCT} is a symmetric matrix due to the symmetry of the correlation matrix R . The elements of MMP_{DCT} depend on ρ only, which normally ranges from 0.4 to 0.75 [43]. By using $\rho = 0.4$, $\rho = 0.6$, and $\rho = 0.75$, we give three theoretical MMP_{DCT} matrixes in Table 3.1.

Table 3.1: Theoretical 4×4 MMP_{DCT} matrixes for $\rho = 0.40$, $\rho = 0.60$, and $\rho = 0.75$

$\rho = 0.40$				$\rho = 0.60$				$\rho = 0.75$			
0.3687	0.1667	0.0675	0.0289	0.5569	0.1572	0.0397	0.0122	0.7079	0.1190	0.0199	0.0050
0.1667	0.0566	0.0165	0.0052	0.1572	0.0198	0.0021	0.0003	0.1190	0.0038	0.0001	0.0000
0.0675	0.0165	0.0034	0.0008	0.0397	0.0021	0.0001	0.0000	0.0199	0.0001	0.0000	0.0000
0.0289	0.0052	0.0008	0.0001	0.0122	0.0003	0.0000	0.0000	0.0050	0.0000	0.0000	0.0000

Some MMP simulations are also conducted by using two video sequences, *Mobile* and *Foreman*, and the resulting MMP for nearly-DCT matrix (2.6), denoted as MMP'_{DCT} , is shown in Table 3.2. As we expected, low frequency components, especially the DC coefficient, has higher MMP than high frequency components. Furthermore, the $MMP'_{DCT}(0,0)$ of *Foreman* is higher than that of *Mobile*, which is due to the higher correlation coefficient between the spatial adjacent residual pixels in *Foreman* sequence.

Table 3.2: MMP'_{DCT} of two test sequences: *Foreman* and *Mobile*

Foreman				Mobile			
0.5908	0.1070	0.0231	0.0023	0.4095	0.1173	0.0664	0.0234
0.1296	0.0427	0.0109	0.0006	0.1341	0.0702	0.0415	0.0094
0.0381	0.0165	0.0053	0.0002	0.0570	0.0330	0.0178	0.0034
0.0198	0.0100	0.0031	0.0001	0.0122	0.0036	0.0012	0.0001

Chapter 3 AZB Detection Algorithm for H.264 INT Using SAD Information

As we know, previous algorithms [36]–[39] divided the 4×4 transformed block into three regions, which is caused by three different scaling factors used in quantization as shown in equation (2.5). However, after examining the MMP matrix, we found there is a strong MMP performance diversity in the $r = 1$ positions so that we proposed to divide them into two groups. The four-group division method is proposed, which is shown in Figure 3.1.

	0	1	2	3
0	G_0	G_1	G_0	G_2
1	G_1	G_3	G_1	G_3
2	G_0	G_1	G_0	G_2
3	G_2	G_3	G_2	G_3

Figure 3.1: Coefficient grouping in a 4×4 transformed block.

For group G_i , the group DCT MMP, $GMMP_{DCT}(i)$, is defined as the sum of all $MMP_{DCT}(u, v)$ in the corresponding group and can be written as

$$GMMP_{DCT}(i) = \sum_{(u,v) \in G_i} MMP_{DCT}(u, v) \quad (3.20)$$

Table 3.3 shows the relationship between $GMMP_{DCT}(i)$ and ρ . Theoretically, for the typical range of ρ , DCT coefficients in group G_0 and G_1 have about 90% chances to hit the magnitude supreme of the whole block. In other words, if we omit the process of

Chapter 3 AZB Detection Algorithm for H.264 INT Using SAD Information

predicting the magnitude supreme of the DCT coefficients in G_2 and G_3 , the resulting type II error (false negative) could be ignored since the magnitude supreme of the DCT coefficients in G_2 or G_3 is unlikely to be used in the AZB detection algorithm.

Table 3.3: $GMMP_{DCT}(i)$ for the typical range of correlation coefficient ρ

ρ	$GMMP_{DCT}(0)$	$GMMP_{DCT}(1)$	$GMMP_{DCT}(2)$	$GMMP_{DCT}(3)$	$GMMP_{DCT}(0) + GMMP_{DCT}(1)$	$GMMP_{DCT}(2) + GMMP_{DCT}(3)$
0.4	0.5071	0.3664	0.0594	0.0672	0.8735	0.1265
0.5	0.5705	0.3498	0.0393	0.0405	0.9203	0.0797
0.6	0.6365	0.3186	0.0245	0.0205	0.9551	0.0449
0.7	0.7084	0.2700	0.0140	0.0076	0.9784	0.0216
0.8	0.7902	0.2012	0.0070	0.0016	0.9914	0.0086

3.4 Efficient AZB detection algorithm

Based on the above MMP analysis, we propose a new AZB detection algorithm, denoted as EAZB. In EAZB, we will consider only the magnitude upper bound of the frequency components in groups G_0 and G_1 .

Using the four-group transformed block division strategy in Figure 3.1, for the coefficients in G_1 , the middle term of equation (3.3) can be expressed as

$$\sum_{x=0}^3 \sum_{y=0}^3 |\phi_{INT}(x, u) \cdot \phi_{INT}(y, v)| \cdot |a(x, y)| = 2 \cdot SAD - S_2 - \delta(u, v), \text{ for } (u, v) \in G_1 \quad (3.21)$$

$$\text{where} \quad \delta(u, v) = \begin{cases} S_1 & \text{for } (u, v) = (0, 1), (2, 1) \\ S_3 & \text{for } (u, v) = (1, 0), (1, 2) \end{cases} \quad (3.22)$$

According to (3.21) and (3.22), the new magnitude upper bound of INT coefficients

Chapter 3 AZB Detection Algorithm for H.264 INT Using SAD Information

in group G_1 can be derived as

$$\left| \tilde{a}_{INT}(u, v) \right|_{upbound} = 2 \cdot SAD - S_2 - \text{MIN}(\delta(u, v)), \text{ for } (u, v) \in G_1 \quad (3.23)$$

Recalling the sufficient condition for zero quantization (3.1), the new sufficient condition for all frequency coefficients of group G_1 to be quantized to zero can be obtained as

$$SAD < T_{EABZ}^{G_1}(Q_P), \text{ where } T_{EABZ}^{G_1}(Q_P) = \frac{1}{2} \cdot T_{INT}(1, Q_P) + \frac{1}{2} \cdot [S_2 + \text{MIN}(S_1, S_3)] \quad (3.24)$$

Moreover, Moon *et al.* showed in [37] that $SAD < T_{INT}(0, Q_P)$ is a sufficient condition for all the INT coefficients in group G_0 to be quantized to zero. With Moon's result, the proposed SAD threshold is:

$$T_{EABZ}(Q_P) = \text{MIN} \{ T_{INT}(0, Q_P), T_{EABZ}^{G_1}(Q_P) \} \quad (3.25)$$

If SAD is smaller than $T_{EABZ}(Q_P)$, the current 4×4 block will be marked as an AZB and the corresponding INT, quantization (Q), inverse-quantization (IQ), inverse-INT (IINT) operations can be dropped.

3.5 Efficient multi-strategy AZB detection algorithm

To achieve the multi-strategy nonzero sub-block detection, Wang *et al.* proposed a Gaussian based detection model (GM) in [40]. As for GM, the INT coefficients are modeled with a zero mean Gaussian distribution and multiple INT strategies are used:

Chapter 3 AZB Detection Algorithm for H.264 INT Using SAD Information

“skip”, “1×1”, “2×2”, “3×3”, “4×4” by making use of multiple SAD thresholds

$(T_{GM}^{Skip}(Q_p), T_{GM}^{1 \times 1}(Q_p), T_{GM}^{2 \times 2}(Q_p), T_{GM}^{3 \times 3}(Q_p))$, which are defined as follows

$$\begin{aligned} T_{GM}^{Skip}(Q_p) &= 0.4493 \cdot T_{INT}(0, Q_p), \quad T_{GM}^{1 \times 1}(Q_p) = 0.4615 \cdot T_{INT}(1, Q_p) \\ T_{GM}^{2 \times 2}(Q_p) &= 1.0329 \cdot T_{INT}(0, Q_p), \quad T_{GM}^{3 \times 3}(Q_p) = 0.8168 \cdot T_{INT}(1, Q_p) \end{aligned} \quad (3.26)$$

with the relationship as

$$T_{GM}^{Skip}(Q_p) < T_{GM}^{1 \times 1}(Q_p) < T_{GM}^{2 \times 2}(Q_p) < T_{GM}^{3 \times 3}(Q_p) \quad (3.27)$$

For example: if $T_{GM}^{1 \times 1}(Q_p) \leq SAD < T_{GM}^{2 \times 2}(Q_p)$, the “2×2” INT strategy will be used and only the 2×2 low frequency INT coefficients will be calculated. Wang further combined GM with his work in [39] to obtain a hybrid all-zero sub-block detection model, denoted as HM [40]. The motivation of his combination is to subtitle $T_{GM}^{Skip}(Q_p)$ for $T_{Wang}(Q_p)$ (3.13) to improve the AZB detection performance of GM. To compare our proposed EAZB to the algorithm in [40], we combine EAZB with GM to obtain a combined model, denoted as EAZB+GM, to show the usefulness and the performance improvement of EAZB in nonzero INT sub-blocks detection algorithm.

The relationship between the quantization coefficient matrix in (2.5) and the definition of

T_{INT} in (3.1) shows that

$$T_{INT}(0, Q_p) > \frac{1}{2} \cdot T_{INT}(1, Q_p) > \frac{1}{4} \cdot T_{INT}(2, Q_p) \quad (3.28)$$

Because $S_i \geq 0$, we can derive the low bound of $T_{EAZB}(Q_p)$ as

Chapter 3 AZB Detection Algorithm for H.264 INT Using SAD Information

$$T_{EABZ}(Q_p) \geq \frac{1}{2} \cdot T_{INT}(1, Q_p) \quad (3.29)$$

With the definition of $T_{GM}^{1 \times 1}(Q_p)$ in (3.26), the relationship between $T_{EABZ}(Q_p)$ and $T_{GM}^{1 \times 1}(Q_p)$ is obtained in following inequality:

$$\frac{T_{EABZ}(Q_p)}{T_{GM}^{1 \times 1}(Q_p)} \geq \frac{\frac{1}{2} \cdot T_{INT}(1, Q_p)}{T_{GM}^{1 \times 1}(Q_p)} = \frac{\frac{1}{2} \cdot T_{INT}(1, Q_p)}{0.4615 \cdot T_{INT}(1, Q_p)} = 1.0834 \quad (3.30)$$

That is to say, $T_{EABZ}(Q_p)$ is always larger than $T_{GM}^{1 \times 1}(Q_p)$, and this guarantees the following relationship:

$$T_{EABZ}(Q_p) > T_{GM}^{1 \times 1}(Q_p) > T_{GM}^{Skip}(Q_p) \quad (3.31)$$

Therefore, when combining EABZ with GM, $T_{GM}^{Skip}(Q_p)$ and $T_{GM}^{1 \times 1}(Q_p)$ can be substituted just by $T_{EABZ}(Q_p)$ such that the “1×1” type INT strategy should be omitted for EABZ+GM.

Figure 3.2: Flowchart of the proposed EABZ+GM.

3.6 Comparison of AZB detection algorithms

The whole procedure of EAZB+GM is presented as a flow chart in Figure 3.2. The SAD value is checked against the SAD thresholds, gradually from low value ($T_{EAZB}(Q_p)$) to high value ($T_{GM}^{4 \times 4}(Q_p)$), in order to find the appropriate INT type for current block.

The performance measurement of AZB detection algorithm is based on three factors: the number of detected AZBs, the PSNR-rate performance, and the computational complexity. We analyze in this section the number of AZBs detected by the detection algorithm, while examining other two factors in the following sections.

In [40], Wang proved that:

$$T_{Wang}(Q_p) \geq T_{Moon}(Q_p) \geq T_{Sousa}(Q_p) > T_{GM}^{Skip}(Q_p) \quad (3.32)$$

Let's define the second term of Wang's threshold in (3.13) as

$$T_{Wang}^{G_1+G_2}(Q_p) = \frac{1}{2} \cdot T_{INT}(1, Q_p) + \frac{1}{2} \cdot \text{MIN} \{S_1 + S_2, S_0 + S_3, S_2 + S_3, S_0 + S_1\} \quad (3.33)$$

because Wang use this term to indicate the magnitude upper bond of the INT coefficients in group G_0 and G_1 .

Since $S_i \geq 0$, the definitions of $T_{EAZB}^{G_1}(Q_p)$ and $T_{Wang}^{G_1+G_2}(Q_p)$ in (3.24) and (3.33) respectively show that

$$T_{EAZB}^{G_1}(Q_p) \geq T_{Wang}^{G_1+G_2}(Q_p) \quad (3.34)$$

Chapter 3 AZB Detection Algorithm for H.264 INT Using SAD Information

Thus, with the derivation of (3.13) and (3.28), the relation between that $T_{EABZ}(Q_p)$ and $T_{Wang}(Q_p)$ can be derived as

$$T_{EABZ}(Q_p) \geq T_{Wang}(Q_p) \quad (3.35)$$

It can also be known that $\sum_{i=0}^3 S_i = 0$ is the sufficient and necessary condition for

$T_{Wang}(Q_p) = T_{Moon}(Q_p) = T_{Sousa}(Q_p)$, but not for $T_{EABZ}(Q_p) = T_{Wang}(Q_p)$. Considering that

$S_i \geq 0$, we have

$$\begin{cases} T_{EABZ}(Q_p) > T_{Wang}(Q_p) = T_{Moon}(Q_p) = T_{Sousa}(Q_p) > T_{GM}^{Skip}(Q_p), & \text{for } \sum_{i=0}^3 S_i = 0 \\ T_{EABZ}(Q_p) \geq T_{Wang}(Q_p) > T_{Moon}(Q_p) > T_{Sousa}(Q_p) > T_{GM}^{Skip}(Q_p), & \text{for } \sum_{i=0}^3 S_i > 0 \end{cases} \quad (3.36)$$

Our simulations also show that in most cases $T_{EABZ}(Q_p)$ is larger than $T_{Su}(Q_p)$. With the derivation in (3.36), we can safely conclude that the proposed EABZ can provide a more aggressive SAD threshold to detect more AZBs than other two-strategy AZB detection algorithms [36]–[39].

For the multi-strategy case, all algorithms (GM [40], HM [40], and proposed EABZ+GM) share same thresholds that split the bound between type “2×2”, “3×3” and “4×4”, while GM uses $T_{GM}^{Skip}(Q_p)$, HM uses $T_{Wang}(Q_p)$ and EABZ+GM uses $T_{EABZ}(Q_p)$ to detect type “skip”. Equation (3.36) shows that the proposed SAD threshold $T_{EABZ}(Q_p)$ is larger or equal to the other two thresholds $T_{Wang}(Q_p)$ and $T_{GM}^{Skip}(Q_p)$. Therefore, EABZ+GM can detect more “skip” type blocks than GM and HM.

3.7 Computational complexity

The computational complexity of different INT strategies (the complexity of INT, Q, IQ, and IINT) is listed in Table 3.4, where ADD, MUL, SFT and CMP denote operations of addition, multiplication, shift and comparison respectively.

Table 3.4: Computational complexity of INT, Q, IQ, IINT for different INT strategies

INT strategy	Computational Complexity			
	ADD	MUL	SHF	CMP
Skip	0	0	0	0
1×1	48	2	34	32
2×2	96	8	52	32
3×3	132	18	64	32
4×4	176	32	96	32

The values in Table 3.4 are calculated based on the fast implementation of INT and IINT [11], as shown in Figure 3.3.

Figure 3.3: Fast implementation of INT and IINT.

Here, we use “3×3” INT strategy as an example to show how the operation number comes out. Figure 3.3 indicates that 7 ADD and 1 SFT are needed to get

Chapter 3 AZB Detection Algorithm for H.264 INT Using SAD Information

$\tilde{a}_{INT}(0), \tilde{a}_{INT}(1), \tilde{a}_{INT}(2)$ from $a_{INT}(0), a_{INT}(1), a_{INT}(2), a_{INT}(3)$; 6 ADD and 1 SFT are needed to get $\hat{a}(0), \hat{a}(1), \hat{a}(2), \hat{a}(3)$ (the reconstructed residual coefficient) from $\hat{a}_{INT}(0), \hat{a}_{INT}(1), \hat{a}_{INT}(2)$. As for quantization, equation (2.3) indicates that 1 ADD, 1 MUL and 1 SFT are needed for one INT coefficient. The inverse-quantization needs 1 MUL and 1 SFT for one INT coefficient. Furthermore, 2 ADD, 2 SFT and 2 CMP are needed to get a reconstructed pixel value and ensure the dynamic range of it. Thus, we can obtain the operation number of “3×3” INT strategy as follows

$$\begin{array}{rcl}
 \text{INT:} & (7 \times \text{ADD} + 1 \times \text{SFT}) \times 7 & \\
 \text{Q:} & (1 \times \text{ADD} + 1 \times \text{MUL} + 1 \times \text{SFT}) \times 9 & \\
 \text{IQ:} & (1 \times \text{MUL} + 1 \times \text{SFT}) \times 9 & \\
 \text{IINT:} & (6 \times \text{ADD} + 1 \times \text{SFT}) \times 7 & \\
 +) & \text{MISC: } (2 \times \text{ADD} + 2 \times \text{SFT} + 2 \times \text{CMP}) \times 16 & \\
 \hline
 \text{Total:} & 132 \times \text{ADD} + 18 \times \text{MUL} + 64 \times \text{SFT} + 32 \times \text{CMP} & (3.37)
 \end{array}$$

The computational complexity and INT strategies of the AZB detection algorithms are listed Table 3.5. For different INT strategies, the AZB detection algorithms share the same amount of ADD, MUL and SFT but require different CMP. It should be noted that additional computations of $s_0, \dots, s_3$ in (3.9) are not required if a modified order is implemented in calculating during SAD.

Table 3.5 shows that, for two-strategy algorithms, the computational complexity of EAZB is similar with Moon [37], while far lower than Su [38] (about 50% reduction) and Wang [39] (about 75% reduction). For multi-strategy algorithms, the computational

Chapter 3 AZB Detection Algorithm for H.264 INT Using SAD Information

complexity of HM [40] can be roughly thought as the sum of Wang [39] and GM [40]. A similar phenomenon is observed in EAZB+GM.

Table 3.5: Computational complexity and the number of INT strategies of AZB detection algorithms

Algorithm	Computational Complexity				INT strategies
	<i>ADD</i>	<i>MUL</i>	<i>SHF</i>	<i>CMP</i>	
Sousa [36]	0	0	0	[1, 1]	2
Moon [37]	1	0	1	[3, 3]	2
Su [38]	3	0	3	[6, 6]	2
Wang [39]	10	0	6	[9, 9]	2
GM [40]	0	0	0	[1, 2, 3, 4, 4]	5
HM [40]	10	0	6	[9, 10, 11, 12, 12]	5
EAZB	2	0	1	[3, 3]	2
EAZB+GM	2	0	1	[3, 4, 5, 5]	4

3.8 Experimental results

Experiments are carried based on the H.264 joint model reference software with common encoding parameters [44]. The test bench consists of three types of sequences: *Foreman* and *Susie* mainly contain low frequency texture; *Container* and *Silent* mainly contain medium frequency texture; *Stefan* and *Mobile* mainly contain high frequency texture. The CIF sequences (100 frames each) are tested with the IPPP frame structure and fixed quantization parameter. The Hadamard transform feature in motion estimation is shut off so that SAD information is available for the AZB detection algorithms. In order to evaluate the detection performance, false rejection rate (FRR) and false acceptance rate (FAR) are defined as:

Chapter 3 AZB Detection Algorithm for H.264 INT Using SAD Information

$$FRR = \frac{N_I}{N_Z} \cdot 100\% \quad , \quad FAR = \frac{N_{II}}{N_{nZ}} \cdot 100\% \quad (3.38)$$

N_I is the number of type I error (false positive), where zero-quantized INT (ZQINT) coefficients are misclassified as non-zero-quantized INT (NZQINT) coefficients; N_{II} is the number of type II error (false negative), where NZQINT coefficients are misclassified as ZQINT. N_Z is the total number of ZQINT coefficients, while N_{nZ} is the total number of NZQINT coefficients.

Table 3.6: Comparison of FRR and FAR for AZB detection algorithms

(A) $Q_p = 22$												
Algorithm	Foreman		Susie		Container		Silent		Stefan		Mobile	
	<i>FRR</i>	<i>FAR</i>	<i>FRR</i>	<i>FAR</i>	<i>FRR</i>	<i>FAR</i>	<i>FRR</i>	<i>FAR</i>	<i>FRR</i>	<i>FAR</i>	<i>FRR</i>	<i>FAR</i>
	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)
Sousa [36]	89.74		96.42		87.60		88.32		81.38		91.27	
Moon [37]	82.67	Lossless	90.39	Lossless	82.35	Lossless	84.77	Lossless	72.77	Lossless	86.38	Lossless
Su [38]	74.04		82.12		76.81		79.44		66.97		81.46	
Wang [39]	67.92		76.79		72.78		75.19		63.79		78.33	
GM [40]	57.42	0.632	65.38	1.119	62.00	0.387	60.88	0.591	61.62	0.080	74.03	0.072
HM [40]	52.70		61.13		58.64		57.82		57.78		71.22	
EAZB	66.04	0.008	75.14	0.012	71.51	0.004	73.71	0.005	62.91	0.002	77.36	0.001
EAZB+GM	52.21	0.632	60.73	1.119	58.30	0.388	57.41	0.590	57.56	0.080	70.98	0.074
(B) $Q_p = 27$												
Algorithm	Foreman		Susie		Container		Silent		Stefan		Mobile	
	<i>FRR</i>	<i>FAR</i>	<i>FRR</i>	<i>FAR</i>	<i>FRR</i>	<i>FAR</i>	<i>FRR</i>	<i>FAR</i>	<i>FRR</i>	<i>FAR</i>	<i>FRR</i>	<i>FAR</i>
	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)
Sousa [36]	71.26		74.73		77.46		81.68		68.83		84.45	
Moon [37]	55.71	Lossless	57.62	Lossless	65.49	Lossless	71.95	Lossless	62.25	Lossless	77.49	Lossless
Su [38]	45.43		45.74		54.53		61.40		57.68		72.08	
Wang [39]	39.65		40.30		49.45		55.24		55.31		69.15	
GM [40]	35.28	0.883	35.82	2.487	41.89	0.526	44.13	0.446	51.36	0.230	64.88	0.189
HM [40]	28.33		28.28		35.68		38.17		48.14		61.49	
EAZB	38.54	0.013	39.52	0.011	48.55	0.005	54.15	0.004	54.87	0.003	68.57	0.002
EAZB+GM	28.05	0.869	28.09	2.489	35.46	0.530	37.89	0.447	48.04	0.223	61.35	0.189

Chapter 3 AZB Detection Algorithm for H.264 INT Using SAD Information

(c) $Q_p = 32$

Algorithm	Foreman		Susie		Container		Silent		Stefan		Mobile	
	<i>FRR</i>	<i>FAR</i>	<i>FRR</i>	<i>FAR</i>	<i>FRR</i>	<i>FAR</i>	<i>FRR</i>	<i>FAR</i>	<i>FRR</i>	<i>FAR</i>	<i>FRR</i>	<i>FAR</i>
	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)
Sousa [36]	47.64	Lossless	46.26	Lossless	52.99	Lossless	64.85	Lossless	62.47	Lossless	76.49	Lossless
Moon [37]	33.97		32.55		40.26		49.95		55.59		69.13	
Su [38]	25.64		22.41		32.04		37.70		49.34		62.87	
Wang [39]	20.59		17.43		27.24		30.24		45.03		58.44	
GM [40]	19.39	1.400	16.78	2.691	23.90	1.229	25.97	0.649	38.28	0.569	51.72	0.646
HM [40]	13.33		10.25		18.04		18.37		34.34		47.68	
EAZB	18.76	0.061	15.81	0.134	25.46	0.048	27.61	0.055	43.23	0.040	56.58	0.021
EAZB+GM	12.88	1.408	9.86	2.699	17.60	1.238	17.68	0.649	33.91	0.570	47.25	0.647

(d) $Q_p = 37$

Algorithm	Foreman		Susie		Container		Silent		Stefan		Mobile	
	<i>FRR</i>	<i>FAR</i>	<i>FRR</i>	<i>FAR</i>	<i>FRR</i>	<i>FAR</i>	<i>FRR</i>	<i>FAR</i>	<i>FRR</i>	<i>FAR</i>	<i>FRR</i>	<i>FAR</i>
	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)
Sousa [36]	29.03	Lossless	24.02	Lossless	33.36	Lossless	42.15	Lossless	55.09	Lossless	67.75	Lossless
Moon [37]	19.20		13.33		24.46		27.09		45.56		58.31	
Su [38]	12.75		6.64		17.68		16.62		36.24		49.00	
Wang [39]	10.50		5.24		14.98		13.11		32.13		44.45	
GM [40]	11.01	1.725	7.46	1.837	13.94	1.410	14.45	0.275	27.40	0.543	37.76	1.137
HM [40]	6.48		2.92		9.46		7.67		22.13		32.41	
EAZB	9.97	0.061	4.94	0.056	14.43	0.042	12.46	0.007	31.17	0.015	43.43	0.033
EAZB+GM	6.35	1.727	2.85	1.837	9.33	1.410	7.50	0.275	21.87	0.540	32.15	1.138

We present the FRR and FAR results in Table 3.6. The results show that, for two-strategy algorithms, EAZB can achieve the lowest FRR, which is caused by its more aggressive threshold as shown in (3.36). With the help of GM, EAZB+GM can achieve the lowest FRR among all algorithms. As for FAR, since the lossless algorithms [36]–[39] have no type II error, their FAR equals to zero. GM and HM share the same FAR because the lossless algorithm in HM [40] will not produce type II error. EAZB can achieve the

Chapter 3 AZB Detection Algorithm for H.264 INT Using SAD Information

lowest FAR among all lossy detection algorithms. This verified that dropping the investigation on low contribution coefficients will only produce minor type II errors. EAZB+GM achieves similar FAR with GM and HM, as a majority of FAR in EAZB+GM is caused by the GM part.

Table 3.7: PSNR and rate degradation of the lossy AZB detection algorithms

Qp	Algorithm	Foreman		Susie		Container		Silent		Stefan		Mobile	
		$\Delta PSNR$	$\Delta Rate$	$\Delta PSNR$	$\Delta Rate$	$\Delta PSNR$	$\Delta Rate$	$\Delta PSNR$	$\Delta Rate$	$\Delta PSNR$	$\Delta Rate$	$\Delta PSNR$	$\Delta Rate$
		(dB)	(%)	(dB)	(%)	(dB)	(%)	(dB)	(%)	(dB)	(%)	(dB)	(%)
22	GM [40]	0.000	+0.01	+0.006	-0.10	0.000	-0.05	-0.002	-0.20	-0.005	-0.17	0.000	0.00
	HM [40]												
	EAZB	+0.006	+0.13	-0.001	-0.14	-0.002	-0.11	-0.002	-0.20	+0.001	-0.02	0.000	+0.02
	EAZB+GM	0.000	+0.01	+0.006	-0.10	0.000	-0.05	-0.002	-0.21	-0.005	-0.17	-0.001	-0.03
27	GM [40]	+0.002	-0.20	-0.006	-0.24	+0.004	+0.27	0.000	+0.11	+0.002	+0.01	-0.005	-1.77
	HM [40]												
	EAZB	0.000	0.00	0.000	0.00	0.000	0.00	0.000	0.00	+0.005	+0.06	+0.002	-1.73
	EAZB+GM	+0.002	-0.22	-0.006	-0.24	+0.004	+0.27	0.000	+0.11	+0.003	-0.01	-0.005	-1.77
32	GM [40]	-0.013	-0.10	+0.006	-0.15	-0.002	+0.06	-0.016	-0.34	+0.005	-0.28	-0.001	+0.10
	HM [40]												
	EAZB	-0.001	-0.06	+0.004	-0.18	0.000	+0.01	-0.001	-0.08	+0.004	-0.06	+0.006	+0.12
	EAZB+GM	-0.013	-0.10	+0.006	-0.15	-0.002	+0.06	-0.014	-0.25	+0.005	-0.28	-0.001	+0.10
37	GM [40]	+0.001	-0.51	-0.002	+0.18	+0.007	+0.15	-0.001	-0.22	+0.005	+0.42	+0.005	-0.36
	HM [40]												
	EAZB	0.000	0.00	0.000	0.00	0.000	0.00	-0.003	+0.01	0.000	+0.07	0.000	-0.29
	EAZB+GM	+0.001	-0.51	-0.002	+0.18	+0.007	+0.15	-0.001	-0.22	+0.002	+0.15	+0.005	-0.36

The PSNR and rate degradation between the lossy algorithms and the original encoder is shown in Table 3.7. It can be observed that the lossy algorithms can achieve a rate-distortion performance close to lossless algorithms. GM and HM share the same value because they have same FAR as shown in Table 3.6. In average, EAZB achieves the lowest rate-distortion performance difference among all lossy algorithms due to its

Chapter 3 AZB Detection Algorithm for H.264 INT Using SAD Information

lowest FAR. EAZB+GM achieves similar RD performance with GM and HM, as they have similar FAR.

The ratio between the overall complexity of the encoding scheme employing the test algorithms (the complexity of INT, Q, IQ, IINT, and the AZB detection algorithm) and the overall complexity of the original H.264 encoder (the complexity of INT, Q, IQ, IINT) is shown in Figure 3.4 for six test sequences. We calculate the complexity as $ADD+3 \times MUL+SHF+CMP$ by using the information in Table 3.4 and Table 3.5. Compared with the original encoder, the encoding schemes with AZB detection algorithm require less complexity, especially for high Q_p . The scheme with EAZB achieves similar overall complexity as the scheme with HM (worse in low Q_p , better in high Q_p) even though HM always has lower FRR. This is due to the low complexity of EAZB as shown in Table 3.5. When EAZB is implemented in multi-strategy, EAZB+GM can always achieve the largest overall complexity reduction among all these algorithms because of its lowest FRR and its low algorithm complexity. On average, the overall complexity ratio obtained by Sousa [36], Moon [37], Su [38], Wang [39], GM [40], HM[40], EAZB, and EAZB+GM are 67.37%, 58.80%, 52.75%, 51.83%, 49.86%, 45.68%, 45.92% and 40.22%, respectively.

Chapter 3 AZB Detection Algorithm for H.264 INT Using SAD Information

(a) *Foreman*

(b) *Susie*

Figure 3.4: Complexity required by the encoding schemes with AZB detection algorithm, compared with the original H.264 encoding scheme.

Chapter 3 AZB Detection Algorithm for H.264 INT Using SAD Information

(c) Container

(d) Silent

Figure 3.4: (Continued) The complexity required by the encoding schemes with AZB detection algorithm, compared with the original H.264 encoding scheme.

Chapter 3 AZB Detection Algorithm for H.264 INT Using SAD Information

(e) *Stefan*

(f) *Mobile*

Figure 3.4: (Continued) The complexity required by the encoding schemes with AZB detection algorithm, compared with the original H.264 encoding scheme.

52

Figure 3.5: Proportion of the complexity of the AZB detection algorithm and the INT, Q, IQ, IINT operations, when HM [40], EAZB and EAZB+GM are implemented for the *Susie* sequence.

For a further comparison between the most recently published algorithm (HM) and our proposed algorithms (EAZB and EAZB+GM), we take the *Susie* sequence as an example to illustrate the in-depth view of the relationship between the amount of the complexity of AZB detection algorithm and the complexity of INT, Q, IQ, IINT operations. Figure 3.5 shows that the algorithm complexity of EAZB or EAZB+GM is far lower than that of HM, which is in line with Table 3.5. As for “INT, Q, IQ, IINT complexity”, the schemes with EAZB+GM need slightly less “INT, Q, IQ, IINT complexity” than the schemes with

Chapter 3 AZB Detection Algorithm for H.264 INT Using SAD Information

HM, because EAZB+GM can detect more “skip” type blocks than HM, which is proved in (3.37). When Q_p becomes higher, the detection algorithm contributes the majority of overall complexity, thus the low algorithm complexity of EAZB and EAZB+GM becomes crucial to the reduction of overall complexity. In other words, the proposed EAZB and EAZB+GM are particularly useful when the percentage of ZQINT coefficients is high.

3.9 Summary

We provide in this chapter a mathematical model to measure contributions of INT coefficients to AZB detection. Accordingly, we propose the EAZB and EAZB+GM strategies to reduce the complexity of the encoding scheme. As comparing with some recently published methods, the analysis and experiment results show that the proposed algorithms can reduce the complexity and, at the same time, maintain a similar video quality.

Chapter 4

AZB Detection Algorithm for H.264 INT Using HT Information

4.1 Motivation

To achieve low-complexity prediction mode decision, Hadamard transform is used for calculation of mode cost in H.264/AVC [16]. However, the SAD based AZB detection algorithms [36]–[40] do not work when HT is enabled and SAD is not available.

To address this issue, the correlation between INT and HT coefficients is studied, the HT thresholds are defined as the scaling factors of INT thresholds, and the Hadamard transformed block is divided into different groups. Different strategy is derived for different INT type and group. An HT information based AZB detection algorithm, namely HTAZBD, is proposed. The new algorithm can achieve a remarkable complexity reduction, compared with a recently published approach.

This chapter is organized as follows: Section 4.2 gives a brief review on one existing HT information based approach. The relationship between INT and HT coefficients is analyzed in Section 4.3. HTAZBD is proposed in Section 4.4. The computational

Chapter 4 AZB Detection Algorithm for H.264 INT Using HT Information

complexity is discussed in Section 4.5. Experimental results and summary are given in Sections 4.6 and 4.7, respectively.

4.2 Review of related work

Recently, Wang *et al.* [45] presented a sum of absolute transformed difference (SATD) based AZB detection algorithm, named as SATDAZBD in this study. The residual signal of HT was modeled by Gaussian distribution and different thresholds were derived based on different types of transform and quantization functions. The transform and quantization computation is efficiently reduced. Nevertheless, Wang's algorithm only uses SATD value to detect AZB, which is a small part of the HT information. However, we observe that by exploiting the correlation between INT and HT coefficients, the performance of AZB detection can be significantly improved.

4.3 Correlation analysis for INT and HT coefficients

In this section, we analyze the relationship between INT and HT to show that the HT coefficients are useful for the estimation of the corresponding INT coefficients.

To facilitate our analysis, we reuse the modeling in Section 3.3 that a 4×4 residual block can be modeled as a random field with variance σ^2 and a separable correlation coefficient ρ . Considering the similarity between DCT, INT and HT, we uniformly

Chapter 4 AZB Detection Algorithm for H.264 INT Using HT Information

model transformed coefficient, $\tilde{a}_{type}(u, v)$, as a zero mean Laplacian distributed variable.

The variance of the (u, v) th transformed coefficient, $\sigma_{type}^2(u, v)$, can be written as

$$\sigma_{type}^2(u, v) = \sigma^2 \cdot \lambda_{type}(u) \cdot \lambda_{type}(v), \text{ where } \lambda_{type} = \text{Diag}(\phi_{type} \times R \times (\phi_{type})^T) \quad (4.1)$$

Here, $\text{Diag}(\)$ represents the diagonal of a matrix as a row vector, and R is the correlation matrix of a column of the incoming residual block, which is defined in (3.17).

Let $\phi_{type}[i]$ donate the i th row vector of the kernel ϕ_{type} , we can rewrite the 4×4 transform kernel ϕ_{type} to be a 16×16 transform kernel φ_{type} as

$$\varphi_{type}[n] = \text{Vec}^T(\phi_{type}^T[v] \otimes \phi_{type}[u]) \quad (4.2)$$

Where $n = 4 \cdot v + u$, $\otimes$ indicate the outer product, $\text{Vec}(\)$ is the vectorization (the $m \times n$ column vector obtained by stacking the columns of the $m \times n$ matrix on top of one another) of a matrix, and T represent the matrix transpose. Then, (2.1) can be rewritten as

$$\tilde{a}_{type}(u, v) = \varphi_{type}[n] \times \text{Vec}(a^T) \quad (4.3)$$

With (4.3), the correlation coefficient between (u, v) th HT coefficient and INT coefficient, namely $\psi_\rho(u, v)$, can be derived as

$$\begin{aligned} \psi_\rho(u, v) &= \frac{\text{Cov}(\tilde{a}_{HT}(u, v), \tilde{a}_{INT}(u, v))}{\sigma_{HT}(u, v) \cdot \sigma_{INT}(u, v)} \\ &= \frac{E(\tilde{a}_{HT}(u, v) \cdot \tilde{a}_{INT}(u, v))}{\sigma_{HT}(u, v) \cdot \sigma_{INT}(u, v)} = \frac{E(\varphi_{HT}[n] \times \text{Vec}(a^T) \cdot \varphi_{INT}[n] \times \text{Vec}(a^T))}{\sigma^2 \cdot \sqrt{\lambda_{HT}(u) \cdot \lambda_{HT}(v)} \cdot \sqrt{\lambda_{INT}(u) \cdot \lambda_{INT}(v)}} \\ &= \frac{\text{Sum}\{(\varphi_{HT}[n] \otimes \varphi_{INT}^T[n]) \circ E(\text{Vec}(a^T) \otimes \text{Vec}(a^T))\}}{\sigma^2 \cdot \sqrt{\lambda_{HT}(u) \cdot \lambda_{HT}(v)} \cdot \sqrt{\lambda_{INT}(u) \cdot \lambda_{INT}(v)}} \end{aligned} \quad (4.4)$$

Chapter 4 AZB Detection Algorithm for H.264 INT Using HT Information

where $Cov(\cdot)$ stands for covariance, $E(\cdot)$ is the mean, $Sum(\cdot)$ is the sum of matrix array elements, and $\circ$ denotes the Hadamard product.

It can be observed that $E\left(\text{Vec}(a^T) \otimes \text{Vec}^T(a^T)\right)$ is the covariance matrix of 4×4 residual block $a(x, y)$ and has the similar format as equation (4.1) so that the following definition can be derived:

$$E\left(\text{Vec}(a^T) \otimes \text{Vec}^T(a^T)\right) = \sigma^2 \cdot \eta_\rho \quad (4.5)$$

where η_ρ is a matrix whose elements depend only on the pixel correlation coefficient ρ . Thus, $\psi_\rho(u, v)$ only relies on ρ and the spatial frequency position (u, v) .

Equation (4.4) can be simplified as follows:

$$\psi_\rho(u, v) = \frac{\text{Sum}\left\{\left(\varphi_{HT}[n] \otimes \varphi_{INT}^T[n]\right) \circ \eta_\rho\right\}}{\sqrt{\lambda_{HT}(u) \cdot \lambda_{HT}(v)} \cdot \sqrt{\lambda_{INT}(u) \cdot \lambda_{INT}(v)}} \quad (4.6)$$

The typical range of ρ , from 0.4 to 0.75 [43], can be used to calculate $\psi_\rho(u, v)$, as shown in Table 4.1.

Table 4.1: Correlation coefficient between INT and HT coefficients
for the typical range of pixel correlation coefficient ρ

ρ	$\psi_\rho(u, v)$					
	$(u, v) \in \left\{\begin{matrix} (0, 0), (0, 2) \\ (2, 0), (2, 2) \end{matrix}\right\}$	$(u, v) \in \left\{\begin{matrix} (0, 1), (1, 0) \\ (1, 2), (2, 1) \end{matrix}\right\}$	$(u, v) = (1, 1)$	$(u, v) \in \left\{\begin{matrix} (0, 3), (2, 3) \\ (3, 0), (3, 2) \end{matrix}\right\}$	$(u, v) \in \{(1, 3), (3, 1)\}$	$(u, v) = (3, 3)$
0.4	1	0.9761	0.9529	0.8939	0.8726	0.7991
0.5	1	0.9798	0.9600	0.8783	0.8606	0.7714
0.6	1	0.9827	0.9658	0.8629	0.8480	0.7446
0.7	1	0.9851	0.9705	0.8484	0.8358	0.7198
0.8	1	0.9871	0.9744	0.8355	0.8247	0.6981

Chapter 4 AZB Detection Algorithm for H.264 INT Using HT Information

It is observed that several spatial frequency positions share the same $\psi_\rho(u, v)$, due to the similarity between INT and HT kernels (2.3). From Table 4.1, we can find that INT coefficients and HT coefficients are highly correlated. The correlation at low spatial frequency positions is very close or equal to 1 and we note that HT coefficients are equivalent to INT coefficients at frequency positions of $(0, 0)$, $(0, 2)$, $(2, 0)$ and $(2, 2)$, due to the unique similarity between these two transforms.

4.4 Efficient AZB detection algorithm by using HT information

The above analysis reveals a linear correlation between HT coefficients and INT coefficients. We thus propose a Hadamard transform based AZB detection algorithm, namely HTAZBD, for early detection of AZBs.

By using the transform kernel and the definition of λ_{type} in (4.1), λ_{INT} and λ_{HT} can be derived as

$$\lambda_{INT} = 2 \cdot \begin{bmatrix} (1+\rho) \cdot (2+\rho+\rho^2) \\ (5+3\rho-4\rho^2-4\rho^3) \\ (1-\rho) \cdot (1+\rho) \cdot (2-\rho) \\ (5-8\rho+4\rho^2-\rho^3) \end{bmatrix}^T, \lambda_{HT} = 2 \cdot \begin{bmatrix} (1+\rho) \cdot (2+\rho+\rho^2) \\ (1-\rho) \cdot (1+\rho) \cdot (2+\rho) \\ (1-\rho) \cdot (1+\rho) \cdot (2-\rho) \\ (1-\rho) \cdot (2-\rho+\rho^2) \end{bmatrix}^T \quad (4.7)$$

For the better understanding of equation (4.7), we use $\lambda_{INT}(0)$ as an example to show the detailed derivation steps.

Chapter 4 AZB Detection Algorithm for H.264 INT Using HT Information

Recalling the definition of ϕ_{INT} in (2.2) and the definition of R in (3.17), their vector multiplication can be obtained as

$$\phi_{INT} \times R = \begin{bmatrix} (1+\rho^3)+(\rho+\rho^2) & (1+\rho)+(\rho+\rho^2) & (1+\rho)+(\rho+\rho^2) & (1+\rho^3)+(\rho+\rho^2) \\ (\rho-\rho^2)+2(1-\rho^3) & (1-\rho)+2(\rho-\rho^2) & -(1-\rho)-2(\rho-\rho^2) & -(\rho-\rho^2)-2(1-\rho^3) \\ (1+\rho^3)-(\rho+\rho^2) & -(1+\rho)+(\rho+\rho^2) & -(1+\rho)+(\rho+\rho^2) & -(1+\rho^3)-(\rho+\rho^2) \\ -2(\rho-\rho^2)+(1-\rho^3) & -2(1-\rho)-(\rho-\rho^2) & 2(1-\rho)+(\rho-\rho^2) & 2(\rho-\rho^2)-(1-\rho^3) \end{bmatrix} \quad (4.8)$$

Thus, $\lambda_{INT}(0)$ can be derived from its definition (4.1) as follows, which is equal to its expression in equation (4.7).

$$\begin{aligned} \lambda_{INT}(0) &= \text{Diag} \left(\phi_{INT} \times R \times (\phi_{INT})^T \right) \\ &= 2 \cdot \left[(1+\rho^3)+(\rho+\rho^2)+(1+\rho)+(\rho+\rho^2) \right] \\ &= 2 \cdot \left[2+3\rho+2\rho^2+\rho^3 \right] \\ &= 2 \cdot \left[(1+\rho) \cdot (2+\rho+\rho^2) \right] \end{aligned} \quad (4.9)$$

Based on the correlation between INT and HT coefficients, the HT thresholds are introduced as the scaling factors of INT thresholds (3.1). The scale factor is defined as:

$$\begin{aligned} \theta_\rho(u, v) &= \frac{E(|\tilde{e}_{HT}(u, v)|)}{E(|\tilde{e}_{INT}(u, v)|)} = \frac{\sigma_{HT}(u, v)}{\sigma_{INT}(u, v)} = \sqrt{\frac{\lambda_{HT}(u) \cdot \lambda_{HT}(v)}{\lambda_{INT}(u) \cdot \lambda_{INT}(v)}} \\ &= \begin{cases} 1 & \text{for } (u, v) \in G_0 \\ \sqrt{\frac{(1-\rho) \cdot (1+\rho) \cdot (2+\rho)}{(5+3\rho-4\rho^2-4\rho^3)}} & \text{for } (u, v) \in G_1 \\ \sqrt{\frac{(1-\rho) \cdot (2-\rho+\rho^2)}{(5-8\rho+4\rho^2-\rho^3)}} & \text{for } (u, v) \in G_2 \\ \frac{(1-\rho) \cdot (1+\rho) \cdot (2+\rho)}{(5+3\rho-4\rho^2-4\rho^3)} & \text{for } (u, v) \in G_3 \end{cases} \end{aligned} \quad (4.10)$$

Chapter 4 AZB Detection Algorithm for H.264 INT Using HT Information

It can be observed that the four-group division method, which is proposed in Chapter 3 (Figure 3.1), is reused here. This is due to the property of λ_{INT} and λ_{HT} in (4.7).

The typical value of ρ is 0.6 [43]. So we have the scale factor.

$$\theta_{\rho=0.6}(u, v) = \begin{cases} 1 & \text{for } (u, v) \in G_0 \\ 0.6084 & \text{for } (u, v) \in G_1 \\ 0.7031 & \text{for } (u, v) \in G_2 \\ 0.3701 & \text{for } (u, v) \in G_3 \end{cases} \quad (4.11)$$

And the threshold against HT coefficients is then

$$T_{HT}(u, v, Q_p) = T_{INT}(r, Q_p) \cdot \theta_{\rho=0.6}(u, v) \quad (4.12)$$

For a specific Q_p , $T_{HT}(u, v, Q_p)$ is a symmetry 4×4 matrix and can be denoted as $T_{HT}(G_i, Q_p)$ to indicate the shared HT threshold in the same group. Based on the HT threshold $T_{HT}(G_i, Q_p)$, HTAZBD determines five INT types: “skip”, “ 1×1 ”, “ 2×2 ”, “ 3×3 ”, “ 4×4 ”. For example: when the type “ 2×2 ” strategy is used, only the 2×2 low frequency INT coefficients are calculated. Since INT coefficients at different frequency positions play different roles in the AZB detection, we define two types of HT threshold. For G_0 , each HT coefficient magnitude is checked against the threshold, since the HT coefficients are equal to the corresponding INT coefficients in G_0 . The coefficients in G_0 are important for the AZB detection algorithm since they contain the DC component. For G_1 , G_2 and G_3 , the group thresholds are employed where the sum of magnitudes of several HT coefficients is compared against these thresholds. If the sum is smaller than

Chapter 4 AZB Detection Algorithm for H.264 INT Using HT Information

the threshold, all the corresponding INT coefficients will be considered to be zero after quantization.

Following values are calculated in our algorithm for the group HT threshold check:

$$\begin{aligned}
 SH_0 &= |\tilde{a}_{HT}(0,1)| + |\tilde{a}_{HT}(1,0)| \\
 SH_1 &= |\tilde{a}_{HT}(1,2)| + |\tilde{a}_{HT}(2,1)| \\
 SH_2 &= |\tilde{a}_{HT}(0,3)| + |\tilde{a}_{HT}(2,3)| + |\tilde{a}_{HT}(3,0)| + |\tilde{a}_{HT}(3,2)| \\
 SH_3 &= |\tilde{a}_{HT}(1,3)| + |\tilde{a}_{HT}(3,1)| + |\tilde{a}_{HT}(3,3)|
 \end{aligned} \tag{4.13}$$

The whole procedure of HTAZBD is presented in Figure 4.1. The sums of absolute HT values are checked against the HT thresholds, gradually from high frequency positions to low frequency positions, in order to find the appropriate INT strategy for current block.

Chapter 4 AZB Detection Algorithm for H.264 INT Using HT Information

Figure 4.1: Flowchart of the proposed HTAZBD.

Chapter 4 AZB Detection Algorithm for H.264 INT Using HT Information

4.5 Computational complexity

The computational complexity, and the number of INT strategies of SATDAZBD [45] and the proposed HTAZBD are listed in Table 4.2. We use the same INT strategies from Chapter 3, and the complexity of these INT strategies are listed in Table 3.4. For different INT strategies, the AZB detection algorithms share the same amount of ADD, MUL and SFT but require different CMP. The numbers of CMP used in the AZB detection algorithms for the INT strategies (“skip”, “1x1”, “2x2”, “3x3”, “4x4”) are shown in Table 4.2. It should be noted that additional computations of SH_i in (4.13) are not required if a modified order is implemented in calculating during SATD. Table 4.2 shows that, HTAZBD requires higher computational complexity than SATDAZBD, because more coefficients need to be checked. However, as shown in our simulation results, HTAZBD outperforms the SATDAZBD in terms of the reduction of the overall computational complexity (the complexity of INT, Q, IQ, IINT, and the AZB detection algorithm), because HTAZBD can detect more AZBs than SATDAZBD.

Table 4.2: Computational complexity and the number of INT strategies for the HT information based AZB detection algorithms

Algorithm	Computational Complexity				INT strategies
	<i>ADD</i>	<i>MUL</i>	<i>SHF</i>	<i>CMP</i>	
SATDAZBD	0	0	0	[1, 2, 3, 4, 4]	5
HTAZBD	0	0	0	[15, 15, 13, 9, 3]	5

Chapter 4 AZB Detection Algorithm for H.264 INT Using HT Information

4.6 Experimental results

Six sequences (*Foreman*, *Susie*, *Container*, *Silent*, *Stefan*, and *Mobile*) are tested on the H.264 JM8.4 reference codec with common encoding parameters [44] and Hadamard transform based mode decision enabled. The CIF sequences (100 frames each) are tested with the IPPP frame structure and fixed quantization parameters.

In order to evaluate the detection performance, the false rejection rate (FRR) and false acceptance rate (FAR) defined in (3.38) are reused.

Table 4.3: Comparison of FRR and FAR for the HT information based AZB detection algorithms

Q_p	Algorithm	Foreman		Susie		Container		Silent		Stefan		Mobile	
		FRR (%)	FAR (%)	FRR (%)	FAR (%)	FRR (%)	FAR (%)	FRR (%)	FAR (%)	FRR (%)	FAR (%)	FRR (%)	FAR (%)
22	SATDAZBD	66.05	0.354	76.55	0.565	69.98	0.339	73.38	0.309	64.31	0.120	78.00	0.083
	HTAZBD	45.68	0.182	65.42	0.119	47.52	0.319	60.24	0.160	51.20	0.246	67.90	0.092
27	SATDAZBD	40.53	0.661	43.35	2.720	47.10	1.528	54.21	0.633	53.73	0.480	68.87	0.246
	HTAZBD	20.84	0.302	25.89	0.360	26.78	0.396	35.07	0.263	38.26	0.522	56.48	0.255
32	SATDAZBD	20.98	1.820	18.51	5.571	26.93	3.428	30.91	1.057	40.55	1.819	56.35	1.010
	HTAZBD	8.60	0.535	6.92	0.868	14.56	0.625	14.09	0.420	22.61	0.842	40.72	0.556
37	SATDAZBD	10.43	3.777	6.93	5.183	15.45	4.118	15.23	1.819	28.15	3.523	41.61	3.059
	HTAZBD	3.47	0.722	1.34	0.607	6.80	0.787	3.95	0.252	10.74	0.777	22.40	0.709

Table 4.3 shows that FAR of HTAZBD is lower than SATDAZBD in most cases. Furthermore, HTAZBD consistently achieves lower FRR than SATDAZBD. This means that HTAZBD can detect more ZQINT and makes less wrong detections than SATDAZBD. This result verifies that based on our correlation analysis between INT and HT, HTAZBD can accurately judge INT coefficients being zero or not-zero, by using the

Chapter 4 AZB Detection Algorithm for H.264 INT Using HT Information

information from HT coefficients. Table 4.3 also shows that the FAR of HTAZBD is higher than SATDAZBD for some seldom cases like the high frequency texture sequence (e.g., *Stefan* and *Mobile*) with low Q_p . In such a situation, the correlation among residual pixels is low and then the magnitude of high frequency components becomes higher than our assumption, so will the type II error.

Table 4.4: PSNR and rate degradation of the HT information based AZB detection algorithms

Q_p	Algorithm	Susie		Susie		Susie		Susie		Susie		Susie	
		$\Delta PSNR$	$\Delta PSNR$	$\Delta PSNR$	$\Delta PSNR$	$\Delta PSNR$	$\Delta PSNR$	$\Delta PSNR$	$\Delta Rate$	$\Delta PSNR$	$\Delta Rate$	$\Delta PSNR$	$\Delta Rate$
		(dB)	(dB)	(dB)	(dB)	(dB)	(dB)	(dB)	(%)	(dB)	(%)	(dB)	(%)
22	SATDAZBD	-0.006	+0.15	-0.002	-0.35	-0.002	+0.10	+0.004	+0.11	-0.006	+0.03	-0.003	-6.81
	HTAZBD	0.000	+0.05	0.000	+0.18	0.003	+0.04	+0.007	+0.02	-0.009	+0.02	-0.004	-6.78
27	SATDAZBD	+0.002	-0.10	+0.012	-0.15	-0.012	-0.13	+0.003	-0.25	-0.007	-0.17	-0.002	-0.06
	HTAZBD	0.000	-0.12	+0.014	-0.35	-0.006	0.00	+0.003	-0.08	-0.015	-0.17	-0.004	-0.12
32	SATDAZBD	-0.018	-0.56	-0.001	+0.16	0.000	+0.20	+0.004	-0.02	+0.003	-0.24	+0.007	+0.03
	HTAZBD	+0.002	-0.33	-0.015	+0.29	+0.010	+0.10	+0.006	+0.16	+0.003	-0.46	+0.004	+0.10
37	SATDAZBD	+0.009	+0.05	0.000	+0.24	-0.005	-0.15	-0.002	-0.48	-0.037	-0.23	-0.010	+0.01
	HTAZBD	+0.005	+0.31	0.000	-0.06	-0.001	+0.10	-0.007	-0.58	-0.003	-0.19	-0.020	-0.31

The PSNR and rate degradation of the test algorithms are shown in Table 4.4. The results show that both SATDAZBD and HTAZBD can achieve similar RD, with negligible performance degradation. It is noted that the FAR is not directly reflected in RD performance degradation. The first reason is that both algorithms achieves very small FAR. The second reason is that a type II error for AZB detection is a tradeoff between the rate and distortion, which sometime results in a better RD performance, such as the experiment result for Susie with $Q_p = 27$ in Table 4.4.

Chapter 4 AZB Detection Algorithm for H.264 INT Using HT Information

(a) Foreman

(b) Susie

Figure 4.2: Complexity required by the encoding schemes with SATDAZBD and HTAZBD, compared with the original H.264/AVC encoding scheme.

Chapter 4 AZB Detection Algorithm for H.264 INT Using HT Information

(c) Container

(d) Silent

Figure 4.2: (continued) Complexity required by the encoding schemes with SATDAZBD and HTAZBD, compared with the original H.264/AVC encoding scheme.

Chapter 4 AZB Detection Algorithm for H.264 INT Using HT Information

Figure 4.2: (continued) Complexity required by the encoding schemes with SATDAZBD and HTAZBD, compared with the original H.264/AVC encoding scheme.

Chapter 4 AZB Detection Algorithm for H.264 INT Using HT Information

The comparison of the complexity of these two test algorithms (the complexity of INT, Q, IQ, IINT, and the AZB detection algorithm) and the overall complexity of the original H.264/AVC encoder (the complexity of INT, Q, IQ, IINT) is shown in Figure. 4.2. We calculate the complexity as $\text{ADD} + 3 \times \text{MUL} + \text{SHF} + \text{CMP}$ by using the information in Table 3.4 and Table 4.2. The figure shows that the encoding schemes with AZB detection algorithms can reduce complexity efficiently. Compared to SATDAZBD, HTAZBD can achieve higher complexity reduction. In average, overall computational complexities of the schemes with SATDAZBD and HTAZBD are 53.09% and 36.83% of the original H.264/AVC encoder, respectively. The improved achievement of HTAZBD is due to its far lower FRR as shown in Table 4.3. Although the complexity of detection in HTAZBD is higher than that of SATDAZBD, as shown in Table 4.2, the lower FFR of HTAZBD in Table 4.3 results in a lower INT, Q, IQ, IINT complexity, so as the lower overall complexity. The computational complexity invested on the HT coefficient checking in HTAZBD can be justified by such overall complexity reduction.

4.7 Summary

In this chapter, we analyze the correlation between HT coefficients and INT coefficients, and then propose an efficient AZB detection algorithm when Hadamard transform is enabled in the H.264 prediction mode decision. As comparing with a recently published method [45], the analysis and experiment results show that the proposed algorithm can efficiently reduce the overall complexity with negligible RD performance loss.

Chapter 5

Low Complexity H.263 to H.264 Inter Frame Transcoding

5.1 Motivation

With so many different video standards (as reviewed in Section 1), which focused on various applications, being developed and used in our daily live, the interoperability between different applications (e.g., Computer, mobile phone, PDA, and TV) and different networks (e.g., wired internet and wireless mobile net) is becoming very important in our unrestrained information society. Transcoding of video contents make such interaction to be possible between different video coding formats, different networks, and different visual quality requirements.

The diversity of transcoding requirements results in four main research areas [46]: format conversion transcoding, spatial resolution reduction transcoding, temporal resolution reduction transcoding and error-resilience transcoding. Various works have been published in these areas to achieve video transcoding with different transmission network capabilities (e.g., bandwidths, bit error-rates, and packet loss-rates) and different terminal capabilities (e.g., computing power and display capability). Feamster [47] proposed an

Chapter 5 Low Complexity H.263 to H.264 Inter Frame Transcoding

MPEG-2 to H.263 transcoder with interlaced to progressive conversion. Lei [48] proposed an H.263 spatial resolution downscaling transcoder with adaptive MV refinement. Youn [49] proposed a temporal resolution reduction transcoder based on forward dominant vector selection method. Some reports on MPEG-2 to H.264 transcoding are also available, such as [50] and [51], but few on the transcoding between H.263 and H.264.

In this chapter, we propose an efficient H.263 to H.264 inter frame transcoding scheme. In Section 5.2 and 5.3, the selection of transcoding architecture and H.264 features are discussed. Section 5.4 details two fast transcoding algorithms, especially focused on the process of motion refinement and mode decision. The experimental results and the summary are given in Section 5.5 and 5.6, respectively.

5.2 Design issue: transcoding architecture selection

Appropriate transcoder architecture is very important for the efficiency of video transcoding. Generally, there are two choices available: cascaded pixel-domain transcoder (CPDT) and transform-domain transcoder (TDT).

In CPDT, the decoder side decompresses the incoming video stream into the spatial domain, and then the encoder side performs recompression to get a new video with different format, different bit rate, different frame rate, or different spatial resolution. An example of H.263 to H.264 CPDT is shown in Figure 5.1. CPDT can achieve the best

Chapter 5 Low Complexity H.263 to H.264 Inter Frame Transcoding

transcoding quality among all the transcoder architectures because of its drift-free property. CPDT is also the most flexible transcoding architecture, as decoder-loop and encoder-loop are totally independent to each other. However, the high complexity and the large buffer size become the main obstacle of the implementation of CPDT for real-time transcoding.

Figure 5.1: An example of H.263 to H.264 cascaded pixel-domain transcoder.

Figure 5.2: An example of H.263 to H.264 transform-domain transcoder.

As for TDT, this architecture requires lower computational complexity because the transform process between spatial domain and transform domain is avoided. The buffer size of TDT is also smaller than that of CPDT as the encoding buffer is united with the

Chapter 5 Low Complexity H.263 to H.264 Inter Frame Transcoding

decoding buffer. An H.263 to H.264 TDT example is shown in Figure 5.2, which is proposed in [52] by using S-transform [53]. However, TDT may work well for intra frame transcoding, but suffers from drift problems for inter frame transcoding. The linearity assumptions on which the derivation is based are not strictly true, which is due to both the clipping functions performed in the video encoder/decoder and the rounding operations performed in the interpolation for fractional-pixel motion compensation. The failed assumptions cause drift [54] in the output stream such that the appeal of TDT is reduced. For the practical application of TDT, the front-encoder may send intra frames regularly to terminate the drift error. However, this compromising method is way too expensive as far more bits is needed for intra frame than inter frame, even with the introduction of new intra prediction mechanism in H.264. Thus, the intra frame refreshing mechanism is undesirable for transcoding, especially with limited bandwidth.

Due to the drift-free property and the flexibility, we select CPDT as the architecture of the proposed transcoder. In order to overcome the high-complexity issue of CPDT, the selection of H.264 features is conducted in Section 5.3 and several fast algorithms are proposed in Section 5.4.

5.3 Design issue: H.264 feature selection

The transcoding process is a tradeoff between speed and video quality. In this section, the PSNR contribution and the computational complexity are evaluated for the H.264

Chapter 5 Low Complexity H.263 to H.264 Inter Frame Transcoding

features and a feature combination is proposed in order to make good use of the compression capability of H.264 while keeping the complexity low.

Under the transcoding point of view, both H.263 and H.264 are hybrid of inter picture prediction to utilize temporal redundancy and transform coding of the remaining signal to reduce spatial redundancy. But there are a great number of differences between these two standards.

Table 5.1: Comparison of the features in H.263 and H.264 Baseline Profile

Parameter	H.263 Baseline	H.264 Baseline
Transform	8×8 DCT	4×4 INT
Max reference buffer	1 frame	Up to 16 frames
Macroblock structure	16×16 pixels based	
Elementary block structure	8×8 pixels	4×4 pixels
MB partition size per MV	16×16	16×16, 16×8, 8×16, 8×8
Sub-partition below 8×8		8×4, 4×8, 4×4
Motion vector resolution	Half-pel	Quarter-pel
Intra prediction in spatial domain		Nine modes for 4×4 luma block; Four modes for 16×16 luma block; One mode for 4×4 chroma block
Inloop deblocking filter		Pixel-based non-linear
Prediction without residual		Skip mode
Mode decision scheme	SAD based	Rate-distortion based

Several H.264 features are the enhanced version of those in H.263, such as the smaller but drift-free transform, the increased number of reference frames, the increased MV accuracy, the more flexible partition size and the RDO oriented mode decision approach. Other H.264 features are newly introduced, such as the inloop deblocking filter and the intra prediction method.

Chapter 5 Low Complexity H.263 to H.264 Inter Frame Transcoding

The incoming H.263 stream can only provide one MV for one 16×16 MB, as shown in Table 5.1. If the very small sub-partition size, like Mode 4×4 , is used in transcoding, a considerable complexity increasing will be introduced by the motion re-estimation or the motion refinement for these small partitions. However, the performance analysis conducted by Joch *et al.* [55] on seven common test sequences has shown that more than 80% bit savings gained by exploiting all possible MB partitions can be obtained by using partitions not smaller than 8×8 . Hence, sub-partition below 8×8 is not used in our proposal.

As for MCP, the experiments in [56] show that if one reference picture is used in H.264 encoding instead of five, the typical PSNR-rate loss could be in 5% range. For sequences with background that remains similar, the penalty could not exceeding 10%. Because H.263 only uses one reference frame, as shown in Table 5.1, the introduction of MCP into our transcoder will result in a large complexity increasing for the MV refinement in multiple reference frames and a double or triple buffer size for the storage of these reference frames. Since the limited gain of MCP cannot justified its high complexity and storage cost especially for the real-time application, only one reference is used in the H.264 encoder part of our proposed transcoder.

As for the accuracy of MV, the experiments in [57] show that the quarter-pel MV refinement can contribute about more than 1 dB PSNR improvement than half-pel MV.

Chapter 5 Low Complexity H.263 to H.264 Inter Frame Transcoding

We decide to apply quarter-pel refinement in our proposal due to its significant PSNR gain with moderate complexity increase.

Table 5.2: H.264 features applied in the proposed transcoder

Parameter	Proposed Transcoder
Transform	4×4 INT
Max reference buffer	1 frame
Elementary block structure	4×4 pixels
MB partition size per MV	16×16, 16×8, 8×16, 8×8
Motion vector resolution	Quarter-pel
Intra prediction in spatial domain	Nine modes for 4×4 luma block; Four modes for 16×16 luma block; One mode for 4×4 chroma block
Prediction without Residual	Skip mode
Mode Decision Method	Rate-distortion based

Based on these observations, we list the main H.264 features to be applied in our proposed transocder in Table 5.2. We try to reserve those parts contributing high for video quality with lower complexity and take out those low efficiency parts. The purpose of such feature selection is to achieve a good quality-complexity tradeoff for real-time transcoding. It can also be noted that the flexibility of CPDT is reflected in our choice of features. For example, the consequence of sub-pel interpolation is that the incoming H.263 stream has to be decoded fully to the pixel-domain. There is no problem for its implementation in CPDT. But if TDT architecture is applied, it is almost impossible to achieve the sub-pel interpolation in transform domain without drift problem.

5.4 Review on motion estimation and inter mode decision procedure

In H.264 reference software JM 8.4 [16], the method of Lagrange multipliers is implemented in the process of motion estimation and inter mode decision. The main steps of motion estimation and inter mode decision are summarized as follow:

a) Motion estimation initial process: The prediction vector of the 16×16 block is used as the center of the spatial search for all inter modes. The SAD values for 4×4 block are pre-calculated for all MVs of the search range and then be used for fast SAD calculation of all larger blocks in b) and c).

b) Motion estimation for mode 16×16, 8×16, and 16×8: Since the tree structured motion compensation is implemented in H.264, different inter modes split the 16×16 MB into different sizes and different numbers of partitions: 16×16 (one partition), 16×8 (two partitions), 8×16 (two partitions), 8×8 (four partitions).

The exhaustive motion search is computed by first minimizing the following equation for all full-pel positions in the search region to get the best full-pel MV for all partitions:

$$J(\mathbf{m}, \lambda_{MOTION}) = SAD_{luma}(s, c(\mathbf{m})) + \lambda_{MOTION} \cdot (R(\mathbf{m} - \mathbf{p})) \quad (5.1)$$

with $\mathbf{m}$ being the MV, $\mathbf{p}$ being the prediction MV, and λ_{MOTION} being the Lagrange multiplier for motion estimation. $SAD_{luma}(s, c(\mathbf{m}))$ represents the luma part SAD between original block s and its predicted block c at the position designated by $\mathbf{m}$.

Chapter 5 Low Complexity H.263 to H.264 Inter Frame Transcoding

$R(\mathbf{m}-\mathbf{p})$ represents the bits used for coding MVs. Then, the half-pel and quarter-pel motion refinement is performed by minimizing the following equation for eight half-pel positions surrounding the best full-pel and eight quarter-pel positions surrounding the best half-pel

$$J(\mathbf{m}, \lambda_{MOTION}) = SATD_{luma}(s, c(\mathbf{m})) + \lambda_{MOTION} \cdot (R(\mathbf{m}-\mathbf{p})) \quad (5.2)$$

c) Motion estimation for mode 8×8 , 4×8 , 8×4 , and 4×4 for each of the four 8×8 sub-MBs: For all 8×8 sub-MBs, the exhaustive motion search is computed by first minimizing equation (5.1) for all full-pel positions and then minimizing equation (5.2) for both the half-pel motion refinement and the quarter-pel motion refinement for all $P8 \times 8$ modes (8×8 , 4×8 , 8×4 , and 4×4). Then, the best mode for each 8×8 sub-MB is chosen by minimizing the Lagrangian function

$$J_{mode}(s, r, \lambda_{MODE}) = SSD_{luma+chroma}(s, r, mode) + \lambda_{MODE} \cdot (R(s, r, mode))$$

$$mode \in [8 \times 8, 8 \times 4, 4 \times 8, 4 \times 4] \quad (5.3)$$

with $SSD_{luma+chroma}(s, r, mode)$ being the sum of squared difference for both luma and chroma parts between the original block s and its reconstruction r , $R(s, r, mode)$ being the number of header bits associated with choosing the specific mode, and λ_{MODE} being the Lagrange multiplier for mode decision.

Chapter 5 Low Complexity H.263 to H.264 Inter Frame Transcoding

d) Using the results of b) and c) to select the best inter mode for the current MB:

Finally, the selection for best inter mode for each MB is conducted by minimizing the Lagrangian function:

$$J_{mode}(s, r, \lambda_{MODE}) = SSD_{luma+chroma}(s, r, mode) + \lambda_{MODE} \cdot (R(s, r, mode))$$

$$mode \in [16 \times 16, 16 \times 8, 8 \times 16, \text{ and } P8 \times 8 \text{ modes}] \quad (5.4)$$

It should be noted that $J_{P8 \times 8}(s, r, \lambda_{MODE})$ does not need to be calculated because it is already available in c).

5.5 Proposed fast algorithms for low complexity transcoding

With the transcoding architecture chosen in Section 5.2 and H.264 feature selection in Section 5.3, the basic structure of our proposed transcoder is set up. Two fast transcoding algorithms, namely “fast skip mode estimation” and “joint motion refinement with inter mode decision”, are proposed in this section to speed up our transcoding structure by making good use of the available information from incoming stream and the transcoding process.

The flow chart of the proposed transcoding procedure for one MB in P frame is shown in Figure 5.3 and it is concluded as follows: If the input H.263 MB is pre-coded in intra mode, the full intra prediction and mode decision procedure will be implemented. Otherwise, skip mode is checked by using the “fast skip mode estimation” method. If

Chapter 5 Low Complexity H.263 to H.264 Inter Frame Transcoding

skip mode is not directly selected, the “joint motion refinement & inter mode decision” scheme will be used to perform inter modes checking to choose the final mode.

Figure 5.3: Flowchart of the proposed H.263 to H.264 inter frame transcoding scheme with the proposed fast algorithms.

5.5.1 Fast skip mode estimation

The skip mode defined in H.264 standard has very strong effect on the compression performance, which is due to its low signaling cost. The proportion of the skip mode MB (the MB with skip mode after mode decision) for three tested sequences is shown in Figure 5.4. It can be observed that the proportion of the skip mode MB is monotonously

Chapter 5 Low Complexity H.263 to H.264 Inter Frame Transcoding

increasing with the quantization parameter Q_p but varying for different kinds of sequences.

Figure 5.4: Proportion of skip mode MB for *Susie*, *Foreman*, and *Mobile*.

As for the low motion or low detail sequences, like *Susie*, up to 90% MBs are encoded in skip mode when Q_p is increased to 40. However, for the high motion or high detail sequences, like *Mobile*, no more than 10% MBs choose skip mode when Q_p is below 25.

Previously, Yin [58] proposed a fast mode decision algorithm to with early-stop strategies. A threshold $T = Nbits \cdot \lambda_{MODE}$ was defined by Yin, where Nbits equals the

Chapter 5 Low Complexity H.263 to H.264 Inter Frame Transcoding

minimum number of bits required for non-skip inter modes. If skip mode RD cost J_{Skip} is smaller than the threshold, skip mode will be directly selected, otherwise the non-skip inter and intra modes will be checked. Her algorithm is good, when skip mode MBs are in the majority, but not efficient for those high detail or high motion sequences, like the *Mobile* sequence.

To address this issue, two thresholds are used in our proposal, T_1 and T_2 , to achieve different strategies for different kinds of sequences: a) skip the process of skip mode estimation for the high detail or high motion sequences; b) choose skip mode directly for the low detail or low motion sequences.

Since there is no bit cost for skip mode, the RD cost of skip mode can be simplified as

$$J_{Skip}(s, r, \lambda_{MODE}) = SSD_{luma+chroma}(s, r, Skip) \quad (5.5)$$

However $SSD_{luma+chroma}$ cannot be used to skip the process of skip mode estimation because it is only available after the process. Since SAD_{luma} is a good indication of $SSD_{luma+chroma}$ and it can be obtained in very low computational cost, we propose to calculate SAD_{luma} before skip mode estimation and set the threshold T_1 on it to achieve the early-stop strategy. Based on our empirical observation, the threshold T_1 can be defined as

$$T_1 = \begin{cases} 2^{(31+Q_p)/6.7}, & \text{for } 0 \leq Q_p < 35 \\ 100 \cdot Q_p - 2584, & \text{for } 35 \leq Q_p < 51 \end{cases} \quad (5.6)$$

Chapter 5 Low Complexity H.263 to H.264 Inter Frame Transcoding

As for another threshold, we propose T_2 on J_{Skip} like Yin's [58] method, but with more aggressive value to obtain more speed-gain. The threshold T_2 is empirically defined as:

$$T_2 = 80 \cdot \lambda_{MODE}, \text{ where } \lambda_{MODE} = 0.85 \cdot 2^{(Q_p - 12)/3}, \text{ for } 0 \leq Q_p \leq 51 \quad (5.7)$$

The process of the proposed fast skip mode estimation is shown in Figure 5.5 and can be concluded as follows: First, the luma part SAD, SAD_{luma} , is calculated. If $SAD_{luma} > T_1$, skip mode estimation process will be skipped and the inter modes will be checked, otherwise continue to check skip mode. After the process of calculating J_{Skip} , if $J_{Skip} < T_2$, skip mode will be selected and the process of checking inter modes will be skipped, otherwise continue to check inter modes.

Figure 5.5: Flowchart of the proposed fast skip mode estimation.

5.5.2 Joint motion refinement & inter mode decision

In video transcoding, the MVs extracted from the incoming bit stream are always reused in order to avoid the high computational complexity of motion. However, the simple MV reuse scheme can introduce considerable quality degradation [59]. On the other hand, the full inter mode decision scheme of H.264 encoding, as we have reviewed in Section 5.4, is very complex and not desirable for real-time implementation. In this section, we propose a fast scheme to perform MV refinement and inter mode decision jointly.

Figure 5.6: The spatial-temporal candidate MVs in current frame and previous frame.

In the proposed scheme, the MVs from the spatial-temporal neighborhood of current block are considered as the initial MV for MV refinement. Since the partition size of the proposed transcoder is not smaller than 8×8 , each transcoded MB contains four H.264

Chapter 5 Low Complexity H.263 to H.264 Inter Frame Transcoding

MVs and one H.263 MV, while the un-transcoded MB only contains one H.263 MV. Considering their spatial-temporal distance to current MB, a candidate MV set, denoted as Ω , is proposed to be formed by three H.263 MVs from the un-transcoded MBs (including current MB) and eight H.264 MVs from the transcoded MBs, which are located in current frame and previous frame. Their relative locations to current MB are shown in Figure 5.6.

In original inter mode decision procedure, ME is conducted by minimizing equation (5.1) and (5.2), and the final mode is selected by minimizing equation (5.4). However, equation (5.4) needs far more complexity than equation (5.1) or (5.2) due to the reconstructed video block r , which can only be got after going through the whole coding processes including INT, Q, IQ, IINT, etc. On the other hand, equation (5.2) just needs Hadamard transform and the result of equation (5.2) in ME is observed to be useful for inter mode decision. Therefore, equation (5.2) is used to achieve fast inter mode decision in the proposal.

The steps of the proposed “joint motion refinement & inter mode decision” scheme are as follows:

- a)** For each 8×8 partition in current MB, the SATD values for MVs in Ω are calculated and stored for fast SATD calculation of the larger partitions in **b)** and **d)**.

Chapter 5 Low Complexity H.263 to H.264 Inter Frame Transcoding

b) Coarse-level ME & Mode decision for Mode 16×16 , 8×16 , and 16×8 are performed by minimizing the Lagrangian function

$$J_{mode}(\mathbf{m}, \lambda_{MOTION}) = SATD_{luma}(s, c(\mathbf{m})) + \lambda_{MOTION} \cdot R(\mathbf{m} - \mathbf{p}), \quad (5.8)$$

$$\mathbf{m} \in \Omega, mode \in [16 \times 16, 8 \times 16, 16 \times 8, 8 \times 8]$$

One Coarse-level best MV is chosen from Ω for each 8×8 partition, based on the results of Coarse-level ME.

c) Quarter-pel MV refinement is performed on the Coarse-level best MV for each 8×8 partition, which means SATD values of 8 further MVs are calculated and stored.

d) Based on all the stored candidate MVs, the refinement MVs and the corresponding SATD values, Fine-level ME is performed for Mode 16×16 , 16×8 , 8×16 , and 8×8 on each MB by minimization equation (5.9)

$$J_{mode}(\mathbf{m}, \lambda_{MOTION}) = SATD_{luma}(s, c(\mathbf{m})) + \lambda_{MOTION} \cdot R(\mathbf{m} - \mathbf{p}),$$

$$\mathbf{m} \in [\Omega, 8 \text{ quarter-pel MVs surrounding Coarse-level best MV}], \quad (5.9)$$

$$mode \in [16 \times 16, 8 \times 16, 16 \times 8, 8 \times 8]$$

f) Fine-level best mode is chosen by

$$\text{FineLevel_BestMode} = \underset{mode \in [16 \times 16, 16 \times 8, 8 \times 16, 8 \times 8]}{\operatorname{argmin}} \{J_{mode}\} \quad (5.10)$$

g) If skip mode is not directly selected by “fast skip mode estimation” algorithm and J_{Skip} is available, the skip-inter mode decision will be conducted by using equation (5.4)

Chapter 5 Low Complexity H.263 to H.264 Inter Frame Transcoding

The ME results are proposed to be reused during inter mode decision, i.e. the high complex RDO method in original inter mode decision (reviewed in Section 5.4) is skipped. Furthermore, the proposed two-level decision module efficiently reuses all the available MV and SATD information to perform MV refinement. Since only quarter-pel refinement is used, not all MVs in Ω are valuable for further refinement. The Coarse-level ME is used to choose such best candidate MV from Ω for MV refinement in step c). And then, the Fine-level ME & Mode decision is the actual stage to choose the best inter mode.

5.6 Experimental results

To evaluate the performance of the proposed methods, our test sequences include three CIF resolution sequences (100 frames each) and three QCIF resolution sequences (250 frames each) pre-coded by using the test model 8 (TMN8) H.263 encoder [60]. The proposed transcoding methods are implemented on the reference software H.264/AVC JM8.4. The bit rates and the frame structure (IPPP) are fixed for fair comparison: the CIF resolution sequences are pre-coded at 3200kbps and transcoded at 1200kbps; the QCIF resolution sequences are pre-coded at 800kbps and transcoded at 300kbps.

Four inter frame transcoding schemes are evaluated: a) a full CPDT (CPDTFULL) with seven inter modes, 17×17 full-pel search, half-pel and quarter-pel refinement, five reference frames, and all intra modes; b) a simplified CPDT (CPDTSMPL) with four

Chapter 5 Low Complexity H.263 to H.264 Inter Frame Transcoding

inter modes (partition size no smaller than 8×8), 17×17 full-pel search, half-pel and quarter-pel refinement, one reference frame, and only intra mode estimation for incoming intra mode MB; c) a low complexity H.263 to H.264 *P* frame transcoder (YIN) proposed in [61]; d) the proposed transcoder (PRO) with four inter modes (partition size no smaller than 8×8), quarter-pel MV refinement, one reference frame, only intra modes estimation for incoming intra mode MB, and the proposed fast algorithms in Section 5.5. The rate-distortion optimization is enabled for all transcoders.

Table 5.3: PSNR results and encoding time
obtained by transcoding H.263 sequences using four transcoding schemes

Sequence	H.263	CPDTFULL		CPDTSMPPL		YIN [58]		PRO	
	PSNR (dB)	PSNR (dB)	Time (s)	PSNR (dB)	Time (s)	PSNR (dB)	Time (s)	PSNR (dB)	Time (s)
Foreman (QCIF)	40.86	38.19	823.138	37.65	135.300	36.87	88.981	37.54	21.834
Mobile (QCIF)	30.73	29.01	824.648	28.95	136.198	27.95	102.063	28.50	21.172
News (QCIF)	45.95	43.17	908.235	42.76	129.455	42.31	102.899	42.67	18.598
Stefan (CIF)	37.19	34.19	1243.497	33.38	213.944	33.20	149.404	33.52	35.837
Flower (CIF)	35.38	31.50	1206.471	30.92	211.616	30.11	148.420	30.73	31.668
M&D (CIF)	45.25	43.83	1242.308	43.57	213.942	43.44	149.415	43.47	36.546

For performance comparison, the PSNR of transcoded video is computed against the original raw video. The complexity is measured based on total encoding time of the H.264 part.

Chapter 5 Low Complexity H.263 to H.264 Inter Frame Transcoding

Table 5.3 shows the PSNR of the pre-coded H.263 stream and the performance of all four transcoders. In average, PRO achieves 97% complexity reduction with 0.58 dB loss when comparing with CPDTFULL, and 84% complexity reduction with 0.13 dB loss when comparing with CPDTSMPL. Considering the same basic features used in PRO and CPDTSMPL, such complexity reduction is due to the proposed fast algorithms and the simplified inter mode decision procedure. Table 5.3 also shows that PRO outperforms YIN with higher PSNR and more than half coding time saving. There are two reasons: First, PRO performs MV refinement by using spatial-temporal information, while YIN only simply reuses the incoming MV of current MB; Second, YIN performs intra mode estimation for incoming inter MBs, which actually results in limited video quality gain with large complexity increasing.

Table 5.4 shows the performance difference between the proposed transcoding schemes with/without fast skip mode estimation. The value “ T_1 in effect” indicates the proportion of MBs who skip the process of skip mode estimation, while the value “ T_2 in effect” indicate the proportion of MBs who choose skip mode directly without checking other modes. The results show that the “fast skip mode estimation” achieves 1-15% reduction in coding time with negligible PSNR loss. In average, T_1 makes 26% MBs perform early-stop before skip mode estimation and T_2 makes 14% MBs choose skip mode directly and skip other modes. The algorithm performs especially well in the *flower* sequence (Figure 5.7), as half image is covered by the high detail follower garden, where

Chapter 5 Low Complexity H.263 to H.264 Inter Frame Transcoding

T_1 works well, and another half image contains only the low detail sky, where T_2 works well.

Figure 5.7: Sample frame from the *Flower* sequence.

Table 5.4: Performance of the proposed scheme with/without fast skip mode estimation

Sequence	PRO without fast skip mode estimation		PRO		T_1 in effect	T_2 in effect
	PSNR (dB)	Time (s)	PSNR (dB)	Time (s)		
Foreman (QCIF)	37.56	22.247	37.54 (−0.02)	21.834 (−1.9%)	36%	5%
Mobile (QCIF)	28.50	21.452	28.50 (0.00)	21.172 (−1.3%)	28%	0%
News (QCIF)	42.68	19.792	42.67 (−0.01)	18.598 (−6.0%)	18%	9%
Stefan (CIF)	33.54	39.044	33.52 (−0.02)	35.837 (−8.2%)	19%	29%
Flower (CIF)	30.74	37.324	30.73 (−0.01)	31.668 (−15.2%)	38%	34%
M&D (CIF)	43.47	38.111	43.47 (0.00)	36.546 (−4.1%)	18%	5%

Chapter 5 Low Complexity H.263 to H.264 Inter Frame Transcoding

Table 5.5: Performance of the proposed scheme with/without quarter-pel refinement

Sequence	PRO without quarter-pel refinement		PRO	
	<i>PSNR</i> (dB)	<i>Time</i> (s)	<i>PSNR</i> (dB)	<i>Time</i> (s)
Foreman (QCIF)	36.73	19.66	37.54 (+0.81)	21.834 (+11.06%)
Mobile (QCIF)	27.58	19.29	28.50 (+0.92)	21.172 (+9.74%)
News (QCIF)	42.12	17.37	42.67 (+0.55)	18.598 (+7.10%)
Stefan (CIF)	32.355	30.20	33.52 (+1.165)	35.837 (+18.68%)
Flower (CIF)	29.725	28.89	30.73 (+1.005)	31.668 (+9.63%)
M&D (CIF)	42.985	33.24	43.47 (+0.485)	36.546 (+9.93%)

The performance difference between the proposed transcoding schemes with/without quarter-pel refinement is shown in Table 5.5. The scheme without quarter-pel refinement will lose 3.4 dB with only about 15% reduction in coding time in average, which indicates the importance of quarter-pel refinement.

5.7 Summary

In this chapter, an early-stop strategy is introduced to reduce the complexity of skip mode process for all kinds of sequences. The spatial-temporal neighboring MVs are used in our novel “joint motion refinement & mode decision” scheme. Experimental results show the performance improvement of the new scheme, which could be very useful in practical H.263 to H.264 real-time transcoding.

Chapter 6

Conclusion and Future Recommendation

6.1 Conclusion

In this thesis, we investigate and develop several fast algorithms for H.264 encoding and transcoding.

In chapter 3, some recent published AZB detection algorithms are reviewed. A novel maximum magnitude probability model is introduced to analyze these algorithms, and an efficient AZB detection algorithm is then proposed by reusing the SAD information. Compared with previous algorithms, the proposed AZB detection algorithm requires far lower complexity in the detection process with an increased number of successful AZB detection. Therefore, the complexity for coding these AZBs can be saved and the overall complexity of RDO will be considerably reduced for H.264 encoding. The performance gain of our proposed AZB detection algorithm is further justified when it is extended to be a multi-strategy AZB detection algorithm.

To address the issue when SAD information is not available before the integer transform, a Hadamard transform information based AZB detection algorithm is proposed in Chapter 4. The relationship between integer transform and Hadamard transform is studied

Chapter 6 Conclusion and Future Recommendation

and an efficient method for reusing Hadamard transform information is implemented in our proposal. Experimental results show that the proposed AZB detection algorithm can achieve about 40% complexity reduction gain over a recently published method, while maintaining a similar RD performance.

A low complexity H.263 to H.264 inter frame transcoding scheme is proposed in Chapter 5. First, the choice of transcoding architecture and the selection of H.264 features are discussed. Then, two fast algorithms, namely “fast skip mode estimation” and “joint motion refinement & inter mode decision”, are implemented in transcoding. Experimental results show that the proposed inter frame transcoder can achieve a significant complexity reduction with a moderate loss of PSNR-rate performance, compared with the original cascaded pixel-domain transcoding scheme.

It can be concluded that the efficient usage of the available information during encoding, (e.g., the SAD information and the Hadamard transform information) and the available information from incoming stream during transcoding (e.g., the spatial-temporal motion information) is a useful method to speed up H.264 encoding and transcoding.

6.2 Future Directions

For further exploration, the proportion of AZB seems to be not only useful for the complexity reduction, but also useful for the rate control in H.264 encoding and transcoding.

Chapter 6 Conclusion and Future Recommendation

Two types of rate and distortion models have been considered in q -domain [62] and ρ -domain [63], respectively. In the q -domain approach, the rate is modeled as a function of the quantization step size and the complexity of the residual signal. In the ρ -domain approach, where ρ is the percentage of zero quantized coefficients, the number of source bits is modeled as a linear function of $(1 - \rho)$. The ρ -domain source rate model is often more accurate than the q -domain source rate model due to the rough estimate of the complexity of residual signals such as MAD. However, ρ -domain approach is not easy to be implemented in H.264 because of a new chicken and egg dilemma: to meet the bit budget for an MB, a quantization parameter should be first determined for the MB by using ρ of the current MB. However, ρ of the current MB is only available after performing integer transform and quantization. With the AZB detection, ρ can be estimated before integer transform, such that the dilemma can be resolved. More work should be done to improve the accuracy of the AZB detection in order to get good ρ estimation and accurate rate estimation.

We observed that the accuracy of the ρ -domain rate model can be enhanced by including the “run” information. The run-level coding is used in H.264/AVC. However, both the ρ -domain approach and the q -domain approach model only the “level” information, no model is available for the “run” information. As AZB detection algorithm can get an estimation of the distribution of zero quantized coefficients, it can also be used

Chapter 6 Conclusion and Future Recommendation

to estimate the “run” information. More work can be done to achieve accurate “run” information estimation, based on the side information of AZB detection process.

Bibliography

Bibliography

- [1] D. P. Kelly, W. van Gestel, T. Hamada, M. Kato, and K. Nakamura, “Blu-ray disc - A versatile format for recording high definition video,” in *Proc. IEEE International Symposium on Consumer Electronics*, pp. 72–73, 2003.
- [2] Skype website. [Online]. Available: <http://www.skype.com>
- [3] Youtube website. [Online]. Available: <http://www.youtube.com>
- [4] ITU-T. Recommendation H.261: Video codec for audiovisual services at px64 Kbit/s, Mar. 1993.
- [5] MPEG-1, “Coding of moving pictures and associated audio,” Tech. Rep., Committee Draft of Standard ISO 11172, 1990.
- [6] MPEG-2, “Information technology-generic coding of moving pictures and associated audio,” Tech. Rep., ISO/IEC 13818-2, Committee Draft, Mar. 1994.
- [7] ITU-T. Recommendation H.263: Video Coding for Low Bit Rate Communication, version 1: Nov. 1995, version 2: Jan. 1998, version 3: Nov. 2000.
- [8] F. Pereira, and T. Ebrahimi, “The MPEG-4 Book”, Prentice Hall, 2002.
- [9] ITU-T. Recommendation H.264 | ISO/IEC 14496-10, “Advanced Video Coding,” 2003.
- [10] S.-K. Kwon, A. Tamhankar, and K. R. Rao, “Overview of H.264/MPEG-4 Part 10,” *Journal of Visual Communication and Image Representation*, vol. 17, no. 2, pp. 186–216, Apr. 2006.

Bibliography

- [11] H. S. Malvar, A. Hallapuro, M. Karczewicz, and L. Kerofsky, "Low-complexity transform and quantization in H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 598–603, July 2003.
- [12] JVT Website. [Online]. Available: <ftp://standards.polycom.com>.
- [13] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1103–1120, Sep. 2007.
- [14] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.
- [15] Y. Su, and M. T. Sun, "Fast multiple reference frame motion estimation for H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 3, pp. 447–452, Mar. 2006.
- [16] K. P. Lim, G. J. Sullivan, and T. Wiegand, "Text Description of Joint Model Reference Encoding Methods and Decoding Concealment Methods," JVT of ISO/IEC MPEG and ITU-T VCEG, JVT-N046, Hong Kong, China, Jan. 2005.
- [17] Y. Yang, and S. S. Hemami, "Generalized rate-distortion optimization for motion-compensated video coders," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 6, pp. 942–955, Sep. 2000.

Bibliography

- [18] H. Wang, and S. Kwong, “A rate-distortion optimization algorithm for rate control in H.264,” in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 1149–1152, Apr. 2007.
- [19] Y.-K. Tu, J.-F. Yang, and M. T. Sun, “Statistical rate-distortion estimation for H.264/AVC coders,” in *Proc. IEEE International Symposium on Circuits and Systems*, pp. 365–368, May 2006.
- [20] E.-H. Yang, and X. Yu, “Rate distortion optimization for H.264 interframe coding: A general framework and algorithms,” *IEEE Trans. Image Processing*, vol. 16, no. 7, pp. 1774–1784, July 2007.
- [21] D. Marpe, H. Schwarz, and T. Wiegand, “Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 620–636, July 2003.
- [22] Shahriar M. Akramullah, Ishfaq Ahmad, and Ming L. Liou, “Optimization of H.263 Video Encoding Using a Single Processor Computer: Performance Tradeoffs and Benchmarking,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 8, pp. 901–915, Aug. 2001.
- [23] A. Peleg, S. Wilkie, and U. Weiser, “Intel MMX for multimedia PCs,” *Commun. ACM*, vol. 40, no. 1, pp. 25–38, Jan. 1997.
- [24] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, “Motion compensated interframe coding for video conferencing,” in *Proc. National Telecommun. Conf.*, vol. 4, pp. C9.6.1–C9.6.5, 1981.

Bibliography

- [25] 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, pp. 438–442, Aug. 1994.
- [26] M. Ghanbari, "The cross-search algorithm for motion estimation," *IEEE Trans. Commun.*, vol. 38, no. 7, pp. 950–953, Jul. 1990.
- [27] S. Zhu and K. K. Ma, "A new diamond search algorithm for fast block matching motion estimation," *IEEE Trans. Image Processing*, vol. 9, no. 2, pp. 287–290, Feb. 2000.
- [28] W. Li and E. Salari, "Successive elimination algorithm for motion estimation," *IEEE Trans. Image Processing*, vol. 4, no. 1, pp. 105–107, Jan. 1995.
- [29] Y.-L. Chan, and W.-C. Siu, "Search strategy for partial distortion elimination in motion estimation," *Electron. Lett.*, vol. 38, no. 23, pp. 1427–1428, Nov. 2002.
- [30] Y.-W. Huang, S.-Y. Chien, B.-Y. Hsieh, and L.-G. Chen, "An efficient and low power architecture design for motion estimation using global elimination algorithm," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Process.*, vol. 3, pp. 3120–3123, May 2002.
- [31] W. I. Choi, B. Jeon, and J. Jeong, "Fast motion estimation with modified diamond search for variable motion block sizes," in *Proc. IEEE Int. Conf. Image Process.*, vol. 2, pp. 371–374, Sep. 2003.
- [32] H.-Y. C. Tourapis and A. M. Tourapis, "Fast motion estimation within the H.264 codec," in *Proc. IEEE Int. Conf. Multimedia Expo*, vol. 3, pp. 517–520, July 2003.

Bibliography

- [33] P. Yang, Y.-W. He, and S.-Q. Yang, "An unisymmetrical-cross multi-resolution motion search algorithm for MPEG-4 AVC/H.264 coding," in *Proc. IEEE Int. Conf. Multimedia Expo*, pp. 531–534, June 2004.
- [34] P. Yin, H.-Y.C. Tourapis, A. M. Tourapis, and J. Boyce, "Fast mode decision and motion estimation for JVT/H.264," in *Proc. IEEE Int. Conf. Image Process*, vol. 3, pp. 853–856, Sep. 2003.
- [35] D. Wu, S. Wu, K. P. Lim, F. Pan, Z. G. Li, and X. Lin, "Block INTER mode decision for fast encoding of H.264," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Process.*, vol. 3, pp. 181–184, May 2004.
- [36] L. A. Sousa, "General method for eliminating redundant computations in video coding," *Electron. Lett.*, vol. 36, no. 4, pp. 306–307, Feb. 2000.
- [37] Y. H. Moon, G. Y. Kim, and J. H. Kim, "An Improved Early Detection Algorithm for All-Zero Blocks in H.264 Video Encoding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 8, pp. 1053–1057, Aug. 2005.
- [38] C.-Y. Su, "An enhanced detection algorithm for all-zero blocks in H.264 video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 52, no. 2, pp. 598–605, May. 2006.
- [39] H. Wang, S. Kwong, and C. W. Kok, "Efficient prediction algorithm of integer DCT coefficients for H.264/AVC optimization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 4, pp. 547–552, Apr. 2006.

Bibliography

- [40] H. Wang and S. Kwong, "Hybrid Model to Detect Zero Quantized DCT Coefficients in H.264," *IEEE Trans. Multimedia*, vol. 9, No. 4, pp. 728–735, June 2007.
- [41] E. Y. Lam and J. Goodman, "A mathematical analysis of the DCT coefficient distributions for images," *IEEE Trans. Image Processing*, vol. 9, no. 10, pp. 1661–1666, Oct. 2000.
- [42] A. K. Jain, *Fundamentals of Digital Image Processing*, Englewood Cliffs, NJ: Prentice-Hall, 1989.
- [43] I. M. Pao and M. T. Sun, "Modeling DCT coefficients for fast video encoding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 4, pp. 608–616, Jun. 1999.
- [44] TK. Tan, G. Sullivan, and W. Thomas, "Recommended Simulation Common Conditions for Coding Efficiency Experiments", ITU-T VCEG, Doc. VCEG-AE010, Jan. 2007.
- [45] H. Wang and S. Kwong, "Prediction of Zero Quantized DCT Coefficients in H.264/AVC Using Hadamard Transform information," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 4, pp. 510–515, Apr. 2008.
- [46] Ishfaq Ahmad, Xiaohui Wei, Yu Sun, and Ya-Qin Zhang, "Video Transcoding: An Overview of Various Techniques and Research Issues," *IEEE Trans. Multimedia*, vol. 7, no. 5, pp. 793–804, Oct. 2005.
- [47] N. Feamster, and S. Wee, "An MPEG-2 to H.263 transcoder," in *Proc. SPIE Conf. Voice, Video Data Communications*, pp. 164–175, Sep. 1999.

Bibliography

- [48] Z. Lei, and N. D. Georganas, "H.263 video transcoding for spatial resolution downscaling," in *Proc. IEEE Int. Conf. Image Process*, pp. 425–430, Apr. 2002.
- [49] 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.
- [50] M. Kucukgoz, and M. T. Sun, "Early-stop and motion vector re-using for MPEG-2 to H.264 transcoding," in *Proc. SPIE-The International Society for Optical Engineering*, vol. 5308, no. 1, pp. 932–936, 2004.
- [51] C. Chen, B.-H. Wu, and H. H. Chen, "MPEG-2 to H.264 transcoding," in *Picture Coding Symp.*, Dec. 2004.
- [52] Y. Su, J. Xin, A. Vetro, and H. Sun, "Efficient MPEG-2 to H.264/AVC Intra Transcoding in Transform-domain," in *Proc. IEEE Internal Symp. Circuits Syst.*, pp. 1234–1237, May 2005.
- [53] J. Xin, A. Vetro, and H. Sun, "Converting DCT Coefficients to H.264/AVC Transform Coefficients," in *Proc. IEEE Pacific-Rim Conference on Multimedia*, vol. 2, pp. 939–946, Nov. 2004.
- [54] D. Lefol, D. Bull, and N. Canagarajah, "Performance evaluation of transcoding algorithms for H.264," *IEEE Trans. Consumer Electronics*, vol. 52, no. 1, pp. 215–222, Feb. 2006.
- [55] A. Joch, F. Kossentini, and P. Nasiopoulos, "A performance analysis of the ITU-T draft H.26L video coding standard," in *12th International Packet Video Workshop*, pp. 290–306, April 2002.

Bibliography

- [56] A. Puri, X. Chen, and A. Luthra, "Video coding using the H.264/MPEG-4 AVC compression standard," *Signal Processing: Image Communication*, vol. 19, no. 9, pp. 793–849, Oct. 2004.
- [57] T. Wedi and H. G. Musmann, "Motion- and aliasing-compensated prediction for hybrid video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 577–586, July 2003.
- [58] P. Yin, H.-Y.T Cheong, A. M. Tourapis, and Boyce. J, "Fast mode decision and motion estimation for JVT/H.264," in *Proc. IEEE Int. Conf. Image Process*, vol. 3, pp. 853–856, Sep. 2003.
- [59] J. Youn, M. T. Sun, and C.-W. Lin, "Motion estimation for high performance transcoding," *IEEE Trans. Consumer Electronics*, vol. 44, no. 3, pp. 649–658, Aug. 1998.
- [60] ITU-T/SG15, "Video Codec Test Model TMN-8," Portland, June 1997.
- [61] J. Bialkowski, M. Menden, M. Barkowsky, K. Illgner and A. Kaup, "A fast H.263 to H.264 inter-frame transcoder with motion vector refinement," *Picture Coding Symposium*, pp. 47–52, 2004.
- [62] T. Chiang and Y.-Q. Zhang, "A new rate control scheme using quadratic rate distortion model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 1, pp. 246–250, Feb. 1997.

Bibliography

- [63] Z. He, and S. K. Mitra, “A Linear Source Model and a Unified Rate Control Algorithm for DCT Video Coding,” IEEE Trans. Circuits Syst. Video Technol., vol. 12, no. 11, pp. 970–982, Nov. 2002.

Publication List

Publication List

Tianxiao Ye, Yap-Peng Tan, and Ping Xue, “Efficient H.264/AVC All-zero Block Detection for Hadamard Transform based Prediction Mode Decision,” to be submitted shortly to *IEEE Trans. Circuits Syst. Video Technol.*

Seong-Ping Chuah, Zhenzhong Chen, Tianxiao Ye, Yap-Peng Tan and Hock-Chuan Chua, “Throughput Adaptation for Scalable Video Multicast in Wireless Networks,” *IEEE International Symposium on Circuits and Systems* 2009, accepted.

Tianxiao Ye, Yap-Peng Tan, and Ping Xue, “Efficient all-zero block detection algorithm for H.264 integer transform,” *IEEE International Symposium on Circuits and Systems*, pp. 3486–3489, May 2008.

Tianxiao Ye, Yap-Peng Tan, and Ping Xue, “A Low Complexity H.263 to H.264 Transcoder,” *IEEE International Symposium on Circuits and Systems*, pp. 5287–5290, May 2006.

Tianxiao Ye, Yap-Peng Tan, and Ping Xue, “Enhanced H.263 to H.264/AVC Video Transcoding with Adaptive Intra-mode Decision,” *International Conference on Information Communications and Signal Processing*, pp. 1130–1134, Dec. 2005.