The Hong Kong Polytechnic University

Department of Electronic and Information Engineering

New Coding Technologies for Videos

Li-Li Wang

A thesis submitted in partial fulfillment of the requirements for

the degree of Doctor of Philosophy

October 2012

CRETIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_______________________________ (Signed)

WANG Lili (Name of student)

Abstract

In order to save transmission bandwidth and storage space, a series of video coding standards have been established during the past two decades, including MPEG-1/2/4, H.261/H.263 and the current H.264/AVC standard. As compared with earlier standards, H.264/AVC has achieved a significant improvement in the rate-distortion performance. That is, it is possible to save bit rate up to 50% of MPEG-2, H.263, or MPEG-4 Part 2 while keeping the video quality about the same. However, the complexity of an encoder based on H.264/AVC standard is thus very high. Therefore, in this thesis, some novel techniques are developed to speed up an encoding process of inter and intra coding in H.264/AVC standard in the initial part of our research work. Experimental results show that the proposed algorithms can achieve a remarkable reduction of encoding complexity with a negligible loss in rate distortion performance compared with those in H.264/AVC. When compared with other algorithms in the literature, the proposed algorithms also give the best performance in terms of both coding efficiency and coding complexity.

With the development of video coding techniques, high definition and ultra-high definition videos are gradually entering our daily life. The demands for the fidelity of the coded videos are higher. The current H.264/AVC standard will soon be unable to efficiently meet the compression demand for transmission or storage at much low bit rate. To develop some new video coding techniques which can give a better rate distortion performance than H.264/AVC standard is increasingly important. Hence this thesis also develops two efficient hierarchical intra prediction algorithms for videos based on H.264/AVC standard. Experimental results show that, as compared to the conventional methods, the proposed algorithm can significantly enhance the compression efficiency of lossless intra coding in H.264/AVC.

However, the H.264/AVC standard may not be still the best in terms of the compression of such larger and higher quality video contents. To overcome this problem, a new generation of video coding standard, which is called High Efficiency Video Coding (VCEG) standard, is being developed by the Joint Collaborative Team-Video Coding (JCT-VC). The objective of HEVC is to achieve a 50% reduction in bitrate compared with the high profile of H.264/AVC standard at the expense of computational complexity but without degrading the image quality. It will be ratified as an international standard in early 2013. Compared with H.264/AVC, many advanced coding tools have been adopted in the HEVC standard. For example, up to 35 modes are available for intra prediction of each prediction unit in HEVC. This can provide more accurate predictions and thereby improve the compression efficiency of intra coding. However, the complexity of an encoder based on the HEVC standard is thereby extremely high. We have found that there is still room for improvement in terms of high intra compression efficiency with less encoding complexity. In the last part of this thesis, we propose an adaptive intra modes skipping algorithm for mode decision and signaling processes for HEVC. The proposed algorithm not only reduces the encoding complexity. Importantly, it also enhances the compression efficiency of intra coding in the HEVC standard.

List of Publications

International Journal Papers

Published or Accepted Journal Papers:

1. Li-Li Wang and Wan-Chi Siu, "Improved Lossless Coding Algorithm in H.264/AVC based on Hierarchical Intra Prediction and Coding Mode Selection", Paper no. 043001-10, Journal of Electronic Imaging, Vol.20 (4), October-December 2011.
2. Li-Li Wang and Wan-Chi Siu, "Improved Algorithm for Detecting Zero-Quantized DCT Coefficients in H.264/AVC", IET Image Processing, The Institution of Engineering and Technology, U.K, pp.1-6, 2013.
3. Li-Li Wang and Wan-Chi Siu, "Novel Adaptive Algorithm for Intra Prediction with Compromised Modes Skipping and Signaling Processes in HEVC", Paper accepted, to be published in IEEE Transactions on Circuits and Systems for Video Technology (CSVT), 2013.

Submitted Journal Papers:

4. Li-Li Wang and Wan-Chi Siu, "Algorithm for Detecting Zero-Quantized DCT coefficients based on Lagrangian Optimization of False Acceptance and Rejection Ratios in H.264/AVC", submitted to Signal Processing: Image Communication, Elsevier Science.

Journal Paper in preparation:

5. Li-Li Wang and Wan-Chi Siu, "Improved Lossless Video Coding based on Adaptive Interpolation Filter and Residual Coding Mode Optimization", to be submitted.

International Conference Papers

6. Li-Li Wang and Wan-Chi Siu, “H.264 Fast Intra Mode Selection Algorithm based on Direction Difference Measure in the Pixel Domain”, Proceedings, pp.1037-1040, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP’2009), Apr. 2009, Taipei, Taiwan.
7. Li-Li Wang and Wan-Chi Siu, “Improved Hybrid Coding Scheme for Intra 4x4 Residual Block Produced by H.264/AVC”, Proceedings, pp.949-952, IEEE International Symposium on Circuits and Systems (ISCAS’2010), 30 May-2 June, 2010, Paris, France.
8. Li-Li Wang and Wan-Chi Siu, “Improved Lossless Coding Algorithm in H.264/AVC based on Hierarchical Intra Prediction”, Proceedings, pp.2053-2056, IEEE International Conference on Image Processing, (ICIP’2011), Sep. 2011, Brussels, Belgium.
9. Li-Li Wang and Wan-Chi Siu, “Improved Hierarchical Intra Prediction based on Adaptive Interpolation Filtering for Lossless Compression”, Paper accepted, to be published in ISCAS’2013, May, 2013, Beijing, China.

Acknowledgements

I would like to take this opportunity to express my sincere gratitude to my supervisor, Prof. W.C. Siu, for his giving me this precious chance to pursue my Ph.D. degree. During the period that I worked on this thesis, Prof. Siu's sustained encouragement, patient guidance and kindness, as well as his academic experience, have contributed to every success of my research. Without his endless support, I cannot finish my Ph.D. study. His hard working style and his immense enthusiasm on research have impressed me deeply. This has inspired me to work hard on the thesis. It is beyond doubt that this will continuously influence my future research and career.

I would again like to express my sincere thanks to Dr. Bonnie Law and Dr. Y.L. Chan for their guidance in our research group meetings. I would also like to thank Prof. Kenneth Lam, Dr. H.K. Cheung, Dr. K.O. Cheng, Mr. W.L. Hui, Mr. Y.H. Kam, Dr. C.H. Fu, Mr. K.W. Hung, Mr. C.S. Wong, Mr. K.Y. Yam, Mr. S.H. Tsang, Mr. C. Zhang, Mr. H. Zhong, H. Wu and all other students in the Centre of Multimedia Signal Processing. The sharing of ideas and experience with them has greatly contributed to make every success of my work. It has been a wonderful time to me in these years to work with them.

Meanwhile, I am glad to express my gratitude to the Department of Electronic and Information Engineering and the Centre of Multimedia Signal Processing for providing me a comfortable working environment. And many thanks to the Hong Kong Polytechnic University for the financial support to my research work.

Above all, special thanks to my family for their great love and constant encouragement at all times. They have given me their unequivocal support throughout, as always, for which my mere expression of thanks likewise does not suffice. Without their understanding and patience, it is impossible for me to complete this research study.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AC	Alternating Current
ADD	Addition
AIF	Adaptive Interpolation Filter
AVC	Advanced Video Coding
AZB	All-Zero-Block
CAVLC	Context Adaptive Variable Length Coding
CIF	Common Intermediate Format
CSS	Coding Scheme Selection
CU	Coding Unit
DC	Direct Current
DCT	Discrete Cosine Transform
DDM	Directional Difference Measure
DDS	Strength of Directional Differences
DPCM	Differential Pulse-Code Modulation
FAR	False Acceptance Ratio
FFO	FAR-FRR Optimization
FLC	Fixed Length Coding
FRR	False Rejection Ratio
HDTV	High-definition television

HEVC	High Efficiency Video Coding
HIP	Hierarchical Intra Prediction
IDCT	Inverse DCT
IVM	Improved Variance based Measure
IQ	Inverse Quantization
JPEG	Joint Picture Experts Group
JVT	Joint Video Team
LCU	Largest CU
MB	Macroblock
MC	Motion Compensation
ME	Motion Estimation
MPM	Most Probable Mode
MUL	Multiplication
OLC	One-Layer Coding
P-frame	Forward Predicted Frame
PSNR	Peak Signal-to-Noise Ratio
PU	Prediction Unit
QCIF	Quarter CIF
QP	Quantization Parameter
QDCT	Quantized DCT
RD	Rate Distortion

RDO	Rate Distortion Optimization
RMD	Rough Mode Decision
RQT	Residual Quadtree
SAD	Sum of Absolute Difference
SATD	Sum of Absolute Transformed Differences
SFT	Shift
SSD	Sum of Squared Differences
TLC	Two-Layer Coding
TU	Transform Unit
VCEG	Video Coding Experts Group
ZQDCT	Zero Quantized DCT

Chapter 1 Introduction

1.1 Latest Video Coding Standards

The H.264/AVC standard has been the representative of the state-of-the-art video coding standard in the past several years. Therefore, it is selected as the platform of this research, and some new techniques have been developed based on it in this thesis. With the development of technologies, high definition and ultrahigh definition videos are gradually entering into our daily life. The current H.264/AVC standard will be unable to efficiently meet the compression demand. A new high efficiency video coding (HEVC) standard is thus being developed. This is also selected as the new platform of our research. Some contributions have also been made based on it in this thesis. In the following section, we give an introduction of the current H.264/AVC and the coming HEVC standards.

1.1.1 Current H.264/AVC standard

The H.264/MPEG-4 Part 10 or AVC is a standard for video compression. It was jointly developed by the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG) [1-4]. It was ratified as an international video coding standard in 2003. Currently, the H.264/AVC is a matured international video coding standard and has many applications. A number of advanced coding technologies [1-4] are available in the H.264/AVC standard. As a result, the compression efficiency achieved by H.264/AVC is very high. Due to the advantage, it is being widely adopted for video conferencing, mobile TV broadcasting, video surveillance and many others. However, the encoding complexity is too high for some real-time applications. It is always desirable to find some low-complexity techniques to speed up the encoding process while keeping a similar coding performance to the H.264/AVC standard.

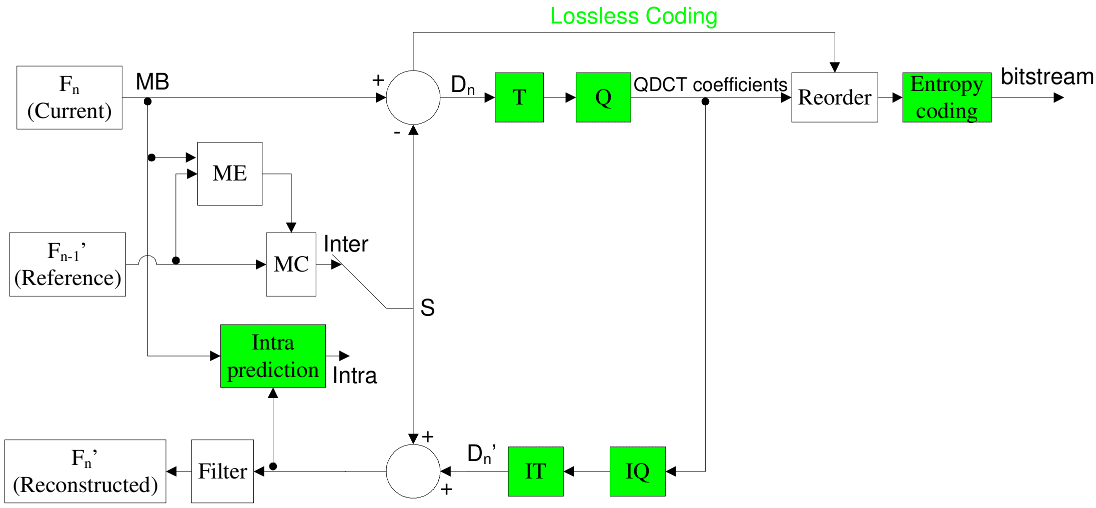


Figure 1.1. Flowchart of an encoder based on H.264/AVC

Figure 1.1 illustrates the flowchart of an encoder based on the reference software JM [5] of H.264/AVC standard. Since H.264/AVC belongs to block-based encoding framework, an input frame F_n is firstly divided into many non-overlapped macroblocks (MBs) with the size of 16x16 for encoding. Each MB is then encoded by intra or inter prediction. In either case, a predicted MB S is formed based on a reconstructed frame. In Intra mode, S is formed from neighboring reconstructed samples in the current frame. In Inter mode, S is formed by motion-compensated prediction from one or more reference frame(s) F'_{n-1}. The difference (residual MB) between original MB and its predicted version is performed by transform (T) and quantization (Q) processes to obtain the quantized DCT (QDCT) coefficients. On the one hand, the QDCT coefficients are encoded into the bit stream. On the other hand, the residual block is reconstructed by performing inverse quantization (IQ) and inverse transform (IT) on the quantized DCT coefficients. The reconstructed residual MB plus the predicted MB create the reconstructed version of an input MB. After deblocking by using a filter for each reconstructed MB, the reconstructed reference frame is obtained from a serious of the filtered-reconstructed MBs. It is then used as the reference for the next frame to do motion estimation and motion compensation.

Our research work based on H.264/AVC in this thesis mainly focuses on two aspects. One is to reduce encoding complexity for real time applications, and the other is to improve compression performance for high resolution videos. All the improved parts are indicated with green color in Figure 1.1.

1.1.2 Coming HEVC standard

With ultrahigh definition video contents entering our daily life, the current H.264/AVC standard may not provide such high compression efficiency to satisfy the transmission requirement of the current internet and broadcasting networks. To overcome this problem, ISO/IEC Moving Picture Experts Group (MPEG) and ITU-T Video Coding Experts Group (VCEG) have formed the Joint Collaborative Team on Video Coding (JCT-VC) entity in April 2010 to develop a new video coding standard, which is called the High Efficiency Video Coding (HEVC) [10, 11]. The objective of HEVC is to substantially improve coding efficiency compared to the H.264/AVC High Profile, i.e. to reduce bitrate requirements by half with comparable image quality, at the expense of increased computational complexity. It is going to be ratified as an international video coding standard in 2013 [10, 11].

Similar to the conventional video coding standards, HEVC still belongs to a block-based hybrid coding framework. The basic block in HEVC is called the largest coding unit (LCU), which can be as large as 64x64 at the present design [10, 11], compared with the MB of 16x16 in the H.264/AVC standard. This large block benefits from encoding of high resolution videos since large smooth regions have the chances to appear in the high resolution picture. In HEVC, each video frame is firstly divided into many non-overlapped LCUs. Each LCU is then recursively split into smaller CUs forming a quadtree representation adapted to the picture. The CU defines a basic unit used for intra and inter prediction. In HEVC, the CU size ranges from 8x8 to 64x64 for Luma component. Figure 1.2 shows an example of the quadtree CU structure in an LCU. Each CU can be further split into Prediction Units (PUs) and Transform Units (TU) in turn. The PU defines a basic unit used for the intra or inter prediction process. The splitting of a CU into PU is different for intra and inter coding. Details can be found in [10, 11]. The TU defines a basic unit for transform and quantization processes. Totally, four TU sizes (4x4, 8x8, 16x16, 32x32) are supported in the current design of HEVC.

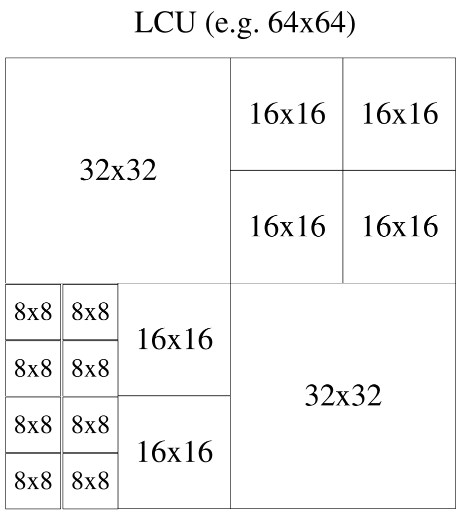


Figure 1.2 An example of splitting LCU into CUs based on a quadtree structure

Compared with the previous coding standards, a number of new coding techniques, such as quad-tree based coding structure, higher bit depth, uniform intra prediction, residual quadtree technique for transform coding, more than one loop filter and so on, are available in the current HEVC [10, 11]. As a result, HEVC has significantly enhanced the compression efficiency than all the existing video coding standards. However, the encoding complexity has thereby increased seriously.

1.2 Literature review

1.2.1 Early detection algorithms of ZQDCT for H.264/AVC

It is well known that the Quantization-based compression which makes use of the discrete cosine transform (DCT) plays an important role in modern video coding standard, such as MPEG-2, MPEG-4 Part 2, H.263, H.264/AVC [1-4]. However, the coding process for an encoder is consequently complicated since DCT, quantization, inverse quantization and inverse DCT (IDCT) are required. The part involving DCT and quantization takes about one fifth of the whole encoding time [6] which cannot be neglected for real-time applications in H.264/AVC. Fortunately, it is observed that many DCT coefficients become zero after quantization. In the case of low bitrate coding, even the whole residual blocks could be zero after quantization. Thus, if the zero quantized DCT (ZQDCT) coefficients can be detected before taking DCT operation, the corresponding DCT, Q, IQ and IDCT can be eliminated. Consequently, considerable computation can be saved for the encoder.

In the past few years, researchers have developed some detection algorithms to look for ZQDCT coefficients in order to save computation. These methods can be classified into two groups, lossless detection method and lossy detection method. For lossless detection method, it is to detect the ZQDCT coefficients without leading to degradation in terms of rate distortion (RD) performance compared with the existing video coding standards. For example, methods in [12-15] are the representatives of lossless detection. These methods are all based on the characteristics of both DCT formula and quantization process. Specifically, Xuan et al.[12] in 1998 first proposed a detection method to determine an all-zero-block (AZB) based on the sum of absolute difference (SAD) of each motion compensated block. There is no additional computation required since the SAD can be obtained during the process of motion estimation. Hence, this algorithm was ever used to reduce significantly the computation for 8x8 DCT in the H.263 video coding process without sacrificing video coding quality. In 2000, Sousa [13] improved Xuan's algorithm and theoretically deduced a more sufficient condition to detect AZB for H.263 encoder. In 2005, Moon and Kim [14] refined the sufficient condition to detect ZQDCT coefficients for the 4x4 integer DCT used in H.264/AVC. In 2008, Jin et al. [15] presented an early detection algorithm of ZQDCT for fast JPEG encoding. To some extent, all these methods can save some computation for an encoder without the degradation of the reconstructed video quality. However, some researchers found that a further saving could be made if a slight loss in the video quality is allowed. It is meaningful for real-time applications. As a consequence, the lossy detection method was developed. It is obvious that the lossy detection method is to detect more ZQDCT coefficients at the expense of a slight degradation of video quality. The related contributions can be seen in papers [16-21]. Among them, a typical method of lossy detection was proposed in [16]. The Gaussian distribution was applied in this paper to study the integer DCT coefficients, and then five types of DCT, Q, IQ and IDCT implementations were suggested to form an early detection method of ZQDCT coefficients in H.264/AVC. In 2010, Chiu et al.[21] proposed an efficient mode decision algorithm for high bitrate coding based on zero-blocks. This is a novel application of the detection method of ZQDCT coefficients.

1.2.2 Fast intra mode decision algorithms for H.264/AVC

In the JVT reference software JM [5], for an intra-MB, the full search algorithm with the rate distortion optimization (RDO) technique [7] is used to examine all the intra prediction modes and to find the best one. Note that the RDO technique for the mode decision is a complex and time-consuming process [22-25]. To reduce the computational complexity it involves, many algorithms have been proposed in recent years. Among them, the way to find the local edge directional information of an MB/block is a popular approach, and has been investigated extensively. There are a number of ways to extract the local edge directional information, such as edge histogram-based method [22], directional filter-based method [23], and gradient-based method [24], etc.

The edge histogram-based method [22] was firstly proposed in 2003. In this algorithm, an edge direction histogram is established from all the pixels in the block by summing up the amplitudes of these pixels with similar directions (angles). According to the histogram, only a small number of intra prediction modes are chosen for RDO calculation. Although this method can reduce the encoding time, the computational load to decide the local edge of a block is still high. Furthermore, the degradation in terms of PSNR is also high based on the experimental results in [22]. Having noted the shortcoming of this algorithm, another fast algorithm based on a filtering technique is proposed in [23]. In this method, each original MB/block is divided into four subblocks firstly, and each subblock will be represented by the average magnitude of its pixels. Thus a 2x2 pseudo block is obtained for edge detection. Five 2x2 filters are then applied to the 2x2 pseudo block to derive the edge strengths of the five corresponding cases. The dominant edge is used as the one with the maximum edge strength. This algorithm can reduce the computational complexity of the dominant edge detection significantly. However, it still has the problem of inaccurate prediction since a 2x2 pseudo block instead of the original block is used to perform the edge detection. It usually does not match the best intra prediction mode with the detected edge. In [24], the authors proposed another algorithm based on local gradient strengths to effectively extract the edge information. This gradient-based algorithm has an accurate detection capacity of candidate prediction modes. Note that it has some drawbacks, such as (i) there is no good reason using the linear interpolation method to compute the gradients of some modes, and (ii) the pseudo-blocks are used for the edge detection of the I16MB type and the Chroma prediction. In addition, a more recent study of intra prediction on fast decision of block size, prediction mode and intra block is found in [25]. In this method, a variance-based method is used to select the block size. The threshold of the variance used in deciding the block size is set to be a constant, which may not be a good arrangement. Note that the variance is fixed for the same MB, however, the encoding mode for the MB can be changed for different QP values. Therefore, the threshold should at least change according to the QP value. On the other hand, for the selected block size, an improved filter-based [23] method is used for mode decision. However, it is found that the filter-based method in [23] has the inaccurate problem, thereby results of inaccuracy may appear in paper [25].

1.2.3 Improved lossless intra coding algorithms for H.264/AVC

Lossless coding technique plays an important role in perfectly preserving valuable information using less storage space. Many approaches have been proposed and are mainly developed for image coding. A good example in the standards is the

JPEG lossless coding [8]. It has many applications, such as medical images, digital archives, digital documentations and so on. In the past few years, some further improvements of lossless image coding studies have been done. One of the representatives is an edge-directed prediction method in [9], which enables the predictor to be adaptive from smooth regions to edge regions. Note that the complexity is usually high for lossless image coding. With the development of video coding standards, high definition videos are gradually entering into our daily life. The demand for higher fidelity of the encoded videos is ever increasing. It is highly possible that either the whole picture or some regions of a picture in a video need to be represented without any loss of fidelity for some demanding applications. Consequently, it is necessary to integrate the lossless coding technique into video coding standards, such as the H.264/AVC coding standard [1-4]. The H.264/AVC standard was originally designed for this purpose of compression. The compression efficiency of video coders can usually be achieved based on lossy coding technique. However, it also supports lossless video coding for some requirements.

The lossless coding method used in H.264/AVC standard can be considered as the state-of-art representative which is applied to the video coding standard. Since there is no transform and quantization process in the lossless video coding, the encoding complexity can be reduced substantially but more bits are required to encode the video sequences. Another problem is that, since the H.264/AVC has a block-based coding structure, samples away from the references cannot be predicted well due to poor correlation. Therefore, further improvement of lossless coding method in H.264/AVC standard is always desirable for the purpose of compression. The third problem is that the entropy coding method CAVLC designed for the quantized DCT coefficients cannot play a good role in coding the residual directly due to the different distributions between the residual and the quantized DCT coefficients.

Recently, researchers have made various studies on the improvement of lossless compression algorithm in the H.264/AVC standard. Lee et al. [26] proposed a lossless intra prediction method based on samplewise DPCM instead of using a block-based approach. This algorithm achieves better performance since near samples are used as the reference, and it is also adopted into a new enhancement project of the H.264/AVC standard. However, the samplewise DPCM method can only be applied to the modes with one sample predictor. As a result, only four modes (modes 0,1,3 and 4) of I4MB/I8MB type, two modes of I16MB type/Chroma prediction can be performed based on the samplewise DPCM method. The performance is thus limited. In [27], a simple interpolation method is used to make prediction. As a result, a samplewise DPCM concept can be easily extended to other prediction modes. The experimental results in [27] and our realization results in Chapter 5 show that further improvement of the lossless compression performance can be achieved compared with the algorithm in [26]. In [28], a two-layer coding algorithm is proposed to improve the lossless coding performance of H.264/AVC standard. In this method, an MB is firstly encoded using lossy coding method in H.264/AVC, and then the distortion between the original MB and the reconstructed MB is also encoded to form the so called D bitstream. The method with two-layer bitstream can be used in applications with multiple clients for different receiving capabilities. However, our experimental results and those from [28] show that this method almost has the same compression rate as the method in H.264/AVC [5], and the coding performance is worse when compared with the method in [26]. Note that this method has a high computational complexity due to two coding processes: lossy coding and lossless coding. Especially for the lossy coding, the complexity is high due to transformation and quantization process. In [29], a pixel-wise interleave prediction algorithm was proposed to perform multidirectional intra prediction as a frame-level coding to improve the lossless intra coding. In this algorithm, four passes corresponding to three sub-frames are available for each frame. The sub-frame in the first pass is encoded by using usual H.264/AVC processing method. The left three sub-frames are encoded by using three directional prediction modes. In [30], we proposed a hierarchical intra prediction (HIP) algorithm making use of intensity gradient measurement in an MB-level to further improve the coding efficiency of lossless intra prediction in H.264/AVC.

1.2.4 Improved intra coding algorithms based on HEVC

Different from intra prediction in H.264/AVC, where at most 9 modes for 8x8 or 4x4 subblock are available in the high profiles [4], the HEVC supports up to 35 modes for uniform intra prediction in each PU [31-33]. The increased modes provide great flexibility to represent complex structures and produce accurate predictions. As a result, the coding efficiency of intra coding method in HEVC is significantly enhanced compared with that in H.264/AVC. However, it also brings a heavy burden to the mode decision process if all the modes are involved in the RDO process. Importantly, more overheads are required to signal the optimal mode for each PU. In order to reduce the complexity of intra coding, a fast mode decision algorithm was proposed in [34] and adopted in the test model HM [35] of HEVC. This algorithm firstly adopts a RMD scheme to select N best candidate modes instead of all the intra prediction modes before the RDO process in HM [35]. Secondly, the best mode is selected from the N candidate modes during the RDO process. The RQT technique [36, 37] is then performed on the residual PU to select the optimal transform structure. This forms a fundamental framework of intra coding process in HM [35]. Based on this framework, many researches have made contributions. On the one hand, fast intra coding algorithms have been proposed to reduce the coding complexity of HEVC. In [38], an early termination strategy based on the intra prediction mode of the corresponding previous-depth prediction unit and block size of current-depth transform unit was proposed for intra prediction. Moreover, the number of candidate modes selected by the RMD scheme is further reduced for the RDO process, and the intra prediction mode of the corresponding previous-depth PU is always included in the candidate modes for the RDO process in [38]. In [39], the authors make full use of the directional information of the neighboring blocks to reduce the number of candidate modes involved in the RDO process. The two fast algorithms can reduce the encoding time by more than 20% for both high efficiency and low complexity conditions at a cost of negligible loss in coding efficiency, as compared with HM. Not only the intra mode decision process can be speeded up, the transform process can also be accelerated. In [40], a merge-and-split decision algorithm was proposed for eliminating the unnecessary transform coding examinations for the RQT process. Furthermore, three early termination schemes were developed for zero-blocks and nonzero-blocks to reduce the unnecessary computation in [40]. This algorithm achieved up to more than 50% encoding time reduction with a slight coding loss when comparing with HM [35]. On the other hand, there are algorithms which can reduce further the coding efficiency of HEVC. For example, authors in [41] proposed an efficient intra mode signaling algorithm for HEVC. This algorithm takes neighboring intra modes into account to obtain a prioritization of the different modes. As a result, entropy coding can be performed with a higher efficiency. About 0.33% to 1% bitrate improvement was achieved with a minimal complexity increase at both encoder and decoder. Note that most of the above algorithms only pay attention to the improvement of one aspect, either RD performance or encoding complexity. When RD performance is improved, the encoding complexity is generally higher, and vice verse. However, we have found that there is still room for improvement in terms of high intra compression efficiency with less encoding complexity.

1.3 Organization of the thesis

The rest of the thesis is organized as follows. In Chapter 2, the basic theory and technologies on transform and quantization, intra prediction, lossless intra coding in H.264/AVC are introduced. Moreover, the intra coding technology in HEVC is also described. In Chapter 3, two efficient detection algorithms of ZQDCT coefficients in H.264/AVC are presented. The first algorithm is based on new patterns for DCT, Q, IQ and IDCT processes and combined with the Gaussian distribution to determine the thresholds for detecting ZQDCT coefficients. The second algorithm is then proposed based on Lagrangian optimization of FAR and FRR in H.264/AVC. In Chapter 4, a fast intra prediction algorithm based on texture characteristics, directional difference measure and early termination for H.264/AVC is proposed. In Chapter 5, two improved lossless intra coding algorithms are suggested to improve the lossless compression efficiency of an encoder based on H.264/AVC for videos. The first algorithm in Chapter 5 is proposed based on hierarchical intra prediction and residual coding mode optimization. The second algorithm is then proposed based on adaptive interpolation filter combined with the residual coding mode optimization. In Chapter 6, an adaptive intra prediction modes skipping algorithm for decision and signaling processes in HEVC is addressed. Finally, a conclusion of this thesis is drawn and some possible future directions of this study is then given in Chapter 7.

Chapter 2 Basic Theories and Technologies

The H.264/AVC is a matured international video coding standard [1-4], which includes a number of advanced coding tools, such as intra prediction, integer transform, lossless intra coding. These technologies make H.264/AVC achieve a significant improvement in rate distortion performance compared with previous standards. In this chapter, we firstly introduce the basic theory of these technologies (in Sections 2.1-2.3), and then present some new techniques adopted in the current design of HEVC (in Section 2.4 of this chapter). Note that the HEVC standard [10, 11] is being developed to achieve better encoding efficiency as compared to the H.264/AVC.

2.1 Transform and quantization techniques in H.264/AVC

2.1.1 Integer DCT in H.264/AVC

Different from the previous video coding standards, such as MPEG-1, MPEG-2 and H.263 where the 8x8 Discrete Cosine Transform (DCT) is the basic transform, the transform used in H.264/AVC for residual data is an integer transform [1-4], which can avoid the operation of floating-point numbers and also eliminate the mismatch problem between the encoding process and decoding process. In the high profile of H.264/AVC, two integer transform sizes (4x4 and 8x8) are supported for residual data after motion-compensated prediction or intra perdition. In our research, we mainly focus on the 4x4 integer DCT in H.264/AVC since 8x8 integer transform is only used in FRExt profiles. The matrix form of 4x4 integer DCT [1-4] can be given by

$$\mathrm{Y} = \left(\mathrm{A}\,\mathbf{XA}^{T}\right) \otimes \mathrm{E}_{f} = \mathrm{W} \otimes \mathrm{E}, \tag{2.1}$$

where matrix **Y** is the output array, matrix **X** is an input array, matrix **A** is an integer DCT kernel, $\otimes$ means term by term multiplication, and **E** is a matrix of scaling factors [1-4]. Matrixes **A** and $\mathbf{E}_{f}$ can be denoted as follows (a=1/2 and $b = \sqrt{2/5}$).

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}, \quad \mathrm{E} = \begin{bmatrix} a^2 & \frac{ab}{2} & a^2 & \frac{ab}{2} \\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4} \\ a^2 & \frac{ab}{2} & a^2 & \frac{ab}{2} \\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4} \end{bmatrix}$$

According to (2.1), $\mathbf{W}=\mathbf{AXA}^{\mathrm{T}}$. Each element of **W**, $\mathbf{W}_{uv}$, denotes an integer DCT coefficient in the position (u,v), which can be expressed as

$$W_{uv} = \sum_{i=0}^{3} \sum_{j=0}^{3} x(i,j) \cdot A(i,u) \cdot A(i,v) \tag{2.2}$$

In (2.2), each element A(m,n) of matrix **A** can be represented by (2.3).

$$A(m,n) = \lceil 2.5 \cdot k(n) \cos((2m+1) \cdot n\pi)/8 \rceil, \tag{2.3}$$

where k(n) is

$$k(n) = \begin{cases} 1/\sqrt{4}, & \text{for } n = 0 \\ 1/\sqrt{2}, & \text{for otherwise} \end{cases}$$

The symbol $\lceil\ \rceil$ in (2.3) is a rounding operator. Note that the calculation of W_{uv} is an integer manipulation process. In the real implementation of an encoder, only matrix **W** instead of matrix **Y** in (2.1) is calculated in transformation process, and $\mathbf{E}_{f}$ is integrated into quantization process. This avoids floating-point numbers operations and thereby eliminates the mismatch problem between the encoding process and decoding process.

2.1.2 Quantization in H.264/AVC

Based on (2.1), each DCT coefficient Y_{uv} in position (u,v) can be calculated as follows:

$$Y_{uv} = W_{uv} \cdot E_{uv} \tag{2.4}$$

In order to quantize the DCT coefficient Y_{uv} in (2.4), the following operation is performed.

$$Z_{uv} = round\left(\frac{Y_{uv}}{Qstep}\right), \tag{2.5}$$

where Z_{uv} is a quantized DCT coefficient, round(·) is a rounding operation, and Qstep denotes quantization step size. In total, 52 values of Qstep ranging from 0 to 51 are supported for Luma component in H.264/AVC, which can be indexed by a quantization parameter (QP) as shown in Table 2.1 [1]. Note that Qstep increases by 12.5% for each increment of 1 in QP. For Chroma component, the maximal value of QP is 39 based on the standard. The wide range of quantization step sizes makes it possible for an encoder based on H.264/AVC to flexibly control the tradeoff between bitrate and quality.

By substituting (2.4) into (2.5), we have

$$Z_{uv} = round\left(\frac{W_{uv} \cdot E_{uv}}{Qstep}\right) \tag{2.6}$$

As we have said that W_{uv} is calculated in transformation process, and E_{uv} is incorporated into quantization process to avoid division operations. In order to integrate E_{uv} into quantization process, E_{uv}/Qstep is defined as follows:

$$\frac{E_{uv}}{Qstep} = \frac{MF_{uv}}{2^{qbits}}, \tag{2.7}$$

where qbits is calculated by (2.8).

$$qbits = 15 + floor(QP/6) \tag{2.8}$$

By substituting (2.7) into (2.6), the factor (E_{uv}/Qstep) is implemented as a multiplication by MF_{uv} and a right-shift operation, thus avoid division operations.

$$Z_{uv} = round\left(W_{uv} \cdot \frac{MF_{uv}}{2^{qbits}}\right), \tag{2.9}$$

where MF_{uv} is calculated by (2.10) based on (2.7).

$$MF_{uv} = E_{uv} / Qstep \times 2^{qbits} \tag{2.10}$$

In the reference software of H.264/AVC [5], a lookup table for MF_{uv} as shown in Table 2.2 is used to save computation and avoid floating-point operations. Parameter MF_{uv} depends on the coefficient position (u, v). The first six values of MF are shown as follows. For QP>5, the MF_{uv} repeats the six values for each increment of 6 in QP.

Table 2.1. Quantization step sizes in H.264/AVC encoder

QP	Qstep	QP	Qstep	QP	Qstep	QP	Qstep
0	0.625	13	2.75	26	13	39	56
1	0.6875	14	3.25	27	14	40	64
2	0.8125	15	3.5	28	16	41	72
3	0.875	16	4	29	18	42	80
4	1	17	4.5	30	20	43	88
5	1.125	18	5	31	22	44	104
6	1.25	19	5.5	32	26	45	112
7	1.375	20	6.5	33	28	46	128
8	1.625	21	7	34	32	47	144
9	1.75	22	8	35	36	48	160
10	2	23	9	36	40	49	176
11	2.25	24	10	37	44	50	208
12	2.5	25	11	38	52	51	224

Table 2.2. Value of MF_{uv} in H.264/AVC

QP	Positions (0, 0), (2, 0), (2, 2), (0, 2)	Positions (1, 1), (1, 3), (3, 1), (3, 3)	Other positions
0	13107	5243	8066
1	11916	4660	7490
2	10082	4194	6554
3	9362	3647	5825
4	8192	3355	5243
5	7282	2893	4559

As a result, the quantized DCT coefficient Z_{uv} in (2.9) can be implemented as (2.11) in the reference software of H.264/AVC [5].

$$Z_{uv} = sign(W_{uv})\left[\left(|W_{uv}| \cdot MF_{uv} + f \cdot 2^{qbits}\right) >> qbits\right], \tag{2.11}$$

where symbol ">>" indicates a binary shift right, and f is 1/3 for intra residual blocks or 1/6 for inter residual blocks in the reference software of H.264/AVC [5].

2.1.3 Conditions for ZQDCT coefficients in H.264/AVC

In this thesis, to search for Zero Quantized DCT coefficients (ZQDCT) is important. It is good that the ZQDCT coefficients be detected before the transformation and quantization processes to save unnecessary calculation involved. In order to make the DCT coefficient Y_{uv} be quantized to zero based on (2.5) in H.264/AVC, it requires

$$|Z_{uv}| < 1 \tag{2.12}$$

By substituting (2.11) into (2.12), we have

$$\frac{|W_{uv}| \cdot MF_{uv} + f \cdot 2^{qbits}}{2^{qbits}} < 1 \tag{2.13}$$

Based on (2.13), we obtain the following condition.

$$|W_{uv}| < \frac{(1-f) \cdot 2^{qbits}}{MF_{uv}} \tag{2.14}$$

If the upper bound of $|W_{uv}|$ is smaller than the right-hand side of (2.14), it means that the DCT coefficient Y_{uv} is quantized into zero. In Section 2.1.4, we analyze two typical methods to find the upper bound of $|W_{uv}|$ to detect ZQDCT coefficients in H.264/AVC as the motivation of our research work.

2.1.4 Analysis of two typical methods for detecting ZQDCT coefficients

The first one [6] was based on a theoretical analysis of the DCT and quantization processes, and the second method [17] was proposed based on two models about the residual pixel values and DCT coefficients, respectively. The reason that we choose these two algorithms is that (i) the former approach can derive a sufficient condition to detection ZQDCT coefficients without leading to the degradation of video quality, and (ii) the latter one can detect more ZQDCT coefficients at the expense of a slight degradation of the video quality. These two methods can be considered as two typical representatives of lossless and lossy detection methods in H.264/AVC, respectively.

To find the upper bound of $|W_{uv}|$, the following inequality operations are performed based on the method in [6].

$$\begin{aligned} |W_{uv}| &= \left|\sum_{i=0}^{3}\sum_{j=0}^{3} x(i,j)\cdot A(i,u)\cdot A(i,v)\right| \\ &\le \sum_{i=0}^{3}\sum_{j=0}^{3} |x(i,j)|\cdot|A(i,u)|\cdot|A(i,v)| \\ &\le \max_{0\le i,j\le 3}\left(|A(i,u)|\cdot|A(i,v)|\right)\cdot\sum_{i=0}^{3}\sum_{j=0}^{3}|x(i,j)| \\ &= \max_{0\le i,j\le 3}\left(|A(i,u)|\cdot|A(i,v)|\right)\cdot SAD \end{aligned}, \tag{2.15}$$

where SAD denotes the sum of absolute difference between the original block and the predicted block. There is no extra calculation required for SAD since it is obtained in the motion estimation process.

By calculation based on (2.3), $\max\limits_{0\le i,j\le 3}\left(\left|A(i,u)\right|\cdot\left|A(i,v)\right|\right)$ is obtained as follows:

$$C_{uv}=\max_{0\le i,j\le 3}\left(\left|A(i,u)\right|\cdot\left|A(i,v)\right|\right)=2^{2-r}\,, \tag{2.16}$$

where r=2-(u%2)-(v%2). Hence the upper bound of $|W_{uv}|$ according to (2.15) is given as follows:

$$\left|W_{uv}\right|_{\max}=C_{uv}\cdot SAD \tag{2.17}$$

By substituting (2.17) into (2.14), we can obtain the sufficient condition to detect ZQDCT coefficients in paper [6] as follows:

$$SAD<\frac{(1-f)\cdot 2^{qbits}}{C_{uv}\cdot MF_{uv}}=\frac{(1-f)\cdot Qstep}{C_{uv}\cdot E_{uv}} \tag{2.18}$$

Therefore, the threshold for the normalized SAD by Qstep in [6] can be derived as (2.19).

$$TN\_MK_{uv}=\frac{1-f}{C_{uv}\cdot E_{uv}} \tag{2.19}$$

In [17], the authors suppose that the residual pixel x at the input of DCT be approximated by a Gaussian distribution with zero mean and variance σ^2 as

$$p(x)=\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{x^2}{2\sigma^2}\right) \tag{2.20}$$

Based on (2.20), the expected value of |x| can be calculated as

$$E(|x|)=\int_{-\infty}^{+\infty}|x|\,p(x)dx=\sqrt{\frac{2}{\pi}}\sigma \tag{2.21}$$

Note that $SAD=\sum_{i=0}^{3}\sum_{j=0}^{3}\left|x(i,j)\right|$. E(|x|) can be approximated as

$$E(|x|)\approx\frac{SAD}{N}\,, \tag{2.22}$$

where N is the number of DCT coefficients (16 for a 4x4 block). By combining (2.21) and (2.22), we have

$$\sigma \approx \sqrt{\frac{\pi}{2}} \frac{SAD}{N} \tag{2.23}$$

Note that the variance, $\sigma_W^2(u,v)$, of the W_{uv} can be written as [46].

$$\sigma_W^2(u,v) = \sigma^2 \left[ARA^T\right]_{u,u} \left[ARA^T\right]_{v,v}, \tag{2.24}$$

where $[\cdot]_{u,u}$ is the $(u,u)^{th}$ component of a matrix, A is given in Section 2.1.1, and R represents the correlation matrix associated with ρ [17, 18]. Parameter ρ is one-step correlation coefficient in the horizontal or vertical direction [17, 18].

By substituting (2.23) into (2.24), we have

$$\sigma_W(u,v) = \sqrt{\frac{\pi\left[ARA^T\right]_{u,u}\left[ARA^T\right]_{v,v}}{2}} \frac{SAD}{N} \tag{2.25}$$

In [17], the Gaussian distribution with variance $\sigma_W^2(u,v)$ as shown in Figure 2.1 is used to model the integer DCT coefficient W_{uv} in H.264/AVC.

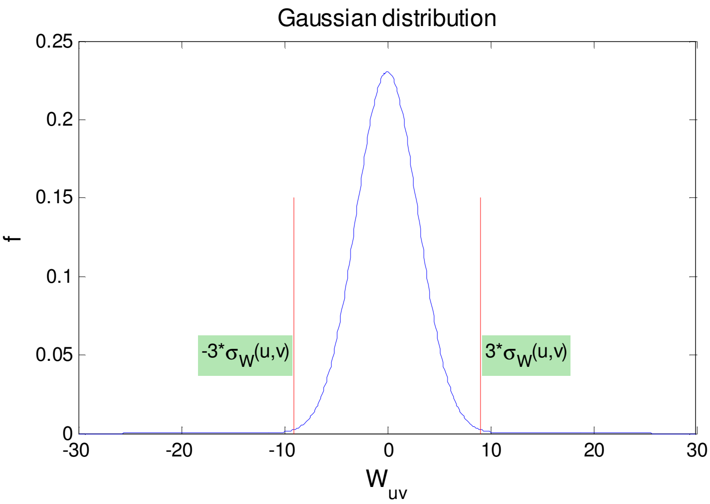


Figure 2.1 Distribution of integer DCT coefficient W_{uv}

Therefore, the maximal possible value of W_{uv} can be expressed by $\sigma_W(u,v)$ with a probability controlled by γ in the following form.

$$|W_{uv}|_{\max} = \gamma\sigma_W(u,v) \tag{2.26}$$

If γ=3, then the probability that the W_{uv} ranges from $-\gamma\sigma_W(u,v)$ to $\gamma\sigma_W(u,v)$ is about 99.73%. By substituting (2.25) and (2.26) into (2.14), we can obtain a different condition from [6] to detect ZQDCT coefficients. By mathematical manipulation, the condition to detect ZQDCT coefficients in [17] can be derived as:

$$SAD < \frac{16\sqrt{2}(1-f)\cdot 2^{qbits}}{\gamma\sqrt{\pi D_{uv}}\cdot MF_{uv}} = \frac{16\sqrt{2}(1-f)\cdot Qstep}{\gamma\sqrt{\pi D_{uv}}\cdot E_{uv}}, \tag{2.27}$$

where $D_{uv}=[ARA^T]_{u,u}[ARA^T]_{v,v}$. Hence the threshold for the normalized SAD by Qstep is

$$TN\_WK_{uv} = \frac{16\sqrt{2}(1-f)}{\gamma\sqrt{\pi D_{uv}}\cdot E_{uv}} \tag{2.28}$$

Note that (2.19) and (2.28) are independent of Qstep. Let us label them as the "normalized thresholds". Table 2.3 shows the normalized threshold for each (u,v) component corresponding to these two methods. In Table 2.3, "\" means that the thresholds are uncertain since the thresholds are block-dependent in [6, 17]. Note that some normalized thresholds obtained by the algorithm in [17] are smaller than those from the algorithm in [6], and others are larger. The former case means that the thresholds used in [17] are still not large enough. As a result, some ZQDCT coefficients cannot be detected. For the latter case, a larger number of ZQDCT coefficients can be detected, which can save much computation during the encoding process. However, some non-ZQDCT coefficients may be incorrectly classified as ZQDCT coefficients based on the algorithm in [17]. This may result in the degradation of the reconstructed video-quality. In other words, the detection capability of one algorithm can be increased at the expense of a slight degradation of the video-quality, or vice versa. Therefore, it is meaningful to find a relationship between the quality of the reconstructed video sequences and the capability of detecting ZQDCT coefficients. If we can find the relationship between them, thresholds for 16 DCT coefficients can then be determined in accordance with the possible requirements of video-quality and encoding complexity.

Table 2.3. Normalized thresholds based on SAD/Qstep

DCT component	Normalized threshold (Inter mode) for SAD/Qstep	
	TN_MK [6]	TN_WK [17]
DC	3.3333	\
AC(1,0)	2.6352	2.4302
AC(2,0)	3.3333	3.4429
AC(3,0)	2.6352	4.3181
AC(0,1)	2.6352	2.4302
AC(1,1)	\	2.4302
AC(2,1)	2.6352	3.4429
AC(3,1)	\	4.3181
AC(0,2)	3.3333	3.4429
AC(1,2)	2.6352	3.4429
AC(2,2)	3.3333	3.4429
AC(3,2)	2.6352	4.3181
AC(0,3)	2.6352	4.3181
AC(1,3)	\	4.3181
AC(2,3)	2.6352	4.3181
AC(3,3)	\	4.3181

Tables 2.4 and 2.5 list the relationship between SAD and the corresponding patterns of DCT, Q, IQ and IDCT for methods in [6] and [17]. Note that the DCT coefficient at a grey position is non-zero, which requires to be computed. In these two tables, thresholds TN_MK_{00} and TN_WK_{00} are block dependent as suggested in [6] and [17], respectively.

Observing Tables 2.4 and 2.5, we can see that there are some differences, which might hint us to give important factors for formulating an efficient detection algorithm of ZQDCT coefficients. First, four modes are available in [6] while five modes are available in [17]. Second, the patterns of DCT, Q, IQ and IDCT in the two methods are also different. Note that when SAD/Qstep is small, such as M1 in Table 2.4, it may belong to M1 or M2 in Table 2.5. According to Table 2.4, four different

DCT coefficients corresponding to a low frequency and three high frequencies require to be calculated. However, according to Table 2.5, the DC coefficient (M1) or 2x2 DCT coefficients (M2) with low frequencies are computed. From experimental results in [17], we know that the method in [17] can detect more ZQDCT coefficients by bringing in a slight video-quality degradation compared with the method in [6], and hence reduce the encoding time of DCT, Q, IQ and IDCT processes.

Table 2.4. Relationship between SAD and pattern for DCT, Q, IQ and IDCT in [6]

Mode	**Condition (Inter)**	**Pattern for DCT, Q, IQ and IDCT**
M0	$\frac{SAD}{Qstep} < TN\_MK_{00}$	Not performed
M1	$TN\_MK_{00} \leq \frac{SAD}{Qstep} < 2$	Y(0,0) Y(1,0) Y(2,0) Y(3,0) Y(0,1) Y(1,1) Y(2,1) Y(3,1) Y(0,2) Y(1,2) Y(2,2) Y(3,2) Y(0,3) Y(1,3) Y(2,3) Y(3,3)
M2	$2.6352 \leq \frac{SAD}{Qstep} < 3.3333$	Y(0,0) Y(1,0) Y(2,0) Y(3,0) Y(0,1) Y(1,1) Y(2,1) Y(3,1) Y(0,2) Y(1,2) Y(2,2) Y(3,2) Y(0,3) Y(1,3) Y(2,3) Y(3,3)
M3	$3.3333 \leq \frac{SAD}{Qstep}$	Y(0,0) Y(1,0) Y(2,0) Y(3,0) Y(0,1) Y(1,1) Y(2,1) Y(3,1) Y(0,2) Y(1,2) Y(2,2) Y(3,2) Y(0,3) Y(1,3) Y(2,3) Y(3,3)

Table 2.5. Relationship between SAD and pattern for DCT, Q, IQ and IDCT in [17]

Mode	**Condition (Inter)**	**Pattern for DCT, Q, IQ and IDCT**
M0	$\frac{SAD}{Qstep} < TN\_WK_{00}$	Not performed
M1	$TN\_WK_{00} \leq \frac{SAD}{Qstep} < 2.4302$	Y(0,0)
M2	$2.4302 \leq \frac{SAD}{Qstep} < 3.4429$	Y(0,0) Y(1,0) Y(0,1) Y(1,1)
M3	$3.4429 \leq \frac{SAD}{Qstep} < 4.3181$	Y(0,0) Y(1,0) Y(2,0) Y(0,1) Y(1,1) Y(2,1) Y(0,2) Y(1,2) Y(2,2)
M4	$4.3181 \leq \frac{SAD}{Qstep}$	Y(0,0) Y(1,0) Y(2,0) Y(3,0) Y(0,1) Y(1,1) Y(2,1) Y(3,1) Y(0,2) Y(1,2) Y(2,2) Y(3,2) Y(0,3) Y(1,3) Y(2,3) Y(3,3)

2.2 Intra coding techniques in H.264/AVC

2.2.1 Intra prediction modes in H.264/AVC

The H.264/AVC introduces directional spatial prediction for intra MB coding. It exploits the spatial correlation between adjacent MBs/blocks. The current MB/block is predicted by the adjacent pixels in the upper and the left MBs/blocks that were decoded earlier. The H.264/AVC allows the intra prediction for both Luma blocks and Chroma blocks. In the FRExt high profiles of the H.264/AVC standard, the Luma intra prediction has three basic types: 16x16 Luma prediction (I16MB type), 8x8 Luma prediction (I8MB type) and 4x4 Luma predicton (I4MB type).

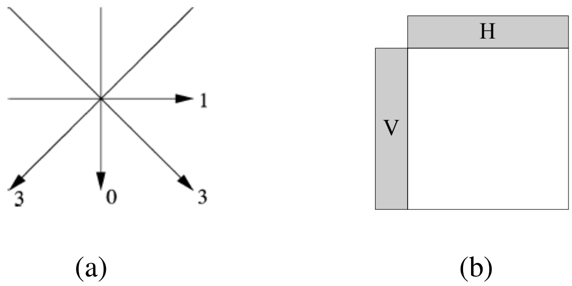


Figure 2.2. (a) Three mathematical prediction directions for I16MB type, (b) Reference pixels for I16MB type

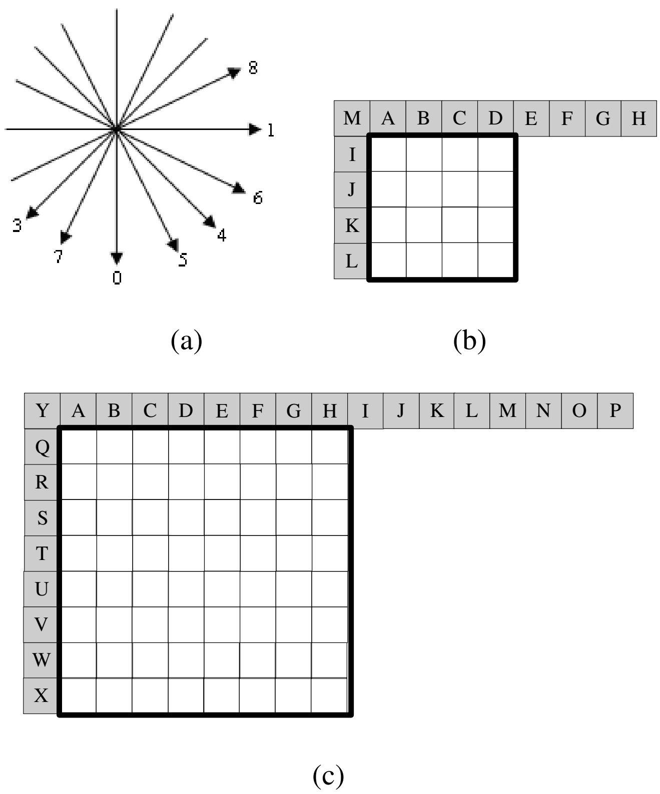


Figure 2.3. (a)Eight prediction directions for I4MB/I8MB type, (b) Reference pixels from A-M for I4MB type, (c) Reference pixels from A-Y for I8MB type

In the I16MB type [1-4], the whole MB is predicted with four possible modes as shown in Figure 2.2. In the I4MB and I8MB types as shown in Figure 2.3 [1-4], each 4x4 or 8x8 block is predicted from spatially neighboring constructed pixels. There are a total of nine prediction modes for each 4x4 or 8x8 Luma block. For the intra prediction of two Chroma components (U and V), there is only one prediction type: 8x8 Chroma prediction (In this thesis, the 4:2:0 Chroma format is mainly used as an example in our realization.). Chroma intra prediction has the same prediction pattern as that of the I16MB type, except its indexing approach and the size of the block which is 8x8. Details about the intra prediction in H.264/AVC can be found in [1-5].

2.2.1 Intra mode selection algorithm based on JM

In the JVT reference software JM12.2 [5], for an intra-MB, the full search algorithm with the RDO technique is used to examine all the intra prediction modes and to find the best one. Figure 2.4 shows the procedure of intra mode selection.

As it can be seen from Figure 2.4, four forms of formulas are applied to compute rate-distortion cost according to different prediction processes.

$$J_{16x16} = \sum_{m=0}^{15} SATD_m^{AC} + SATD^{DC} \tag{2.29}$$

$$J_{8x8} = SSE_{8x8}(s,c,mode \mid QP) + \lambda_{mode} \cdot R(s,c,mode \mid QP) \tag{2.30}$$

$$J_{4x4} = SSE_{4x4}(s,c,mode \mid QP) + \lambda_{mode} \cdot R(s,c,mode \mid QP) \tag{2.31}$$

$$J = SSE_{16x16}(luma,type) + SSE_{8x8}(chroma,mode) + \lambda_{mode} \cdot \max(0.5, R(s,c,mode \mid QP)) \tag{2.32}$$

Equations (2.29)-(2.31) are used to measure the costs of a block encoded as I16MB, I8MB and I4MB, respectively. In (2.29), $SATD_m^{ac}$ represents the sum of 15 absolute AC coefficients which come from the Hadamard transform of the differences between the pixel values of the mth 4x4 block and that of the predicted 4x4 block. $SATD^{dc}$ stands for the sum of absolute Hadamard-transformed coefficients of the 16 DC terms divided by 4. The 16 DC terms come from each DC component of 16 4x4-Hadamard transforms of an MB. In (2.30) and (2.31), SSE represents the sum of the squared errors between the original Luma block s and its reconstructed

Luma block c. QP is the quantization parameter. λ_{mode} is the Lagrange multiplier for mode decision. *mode* represents one of the nine intra prediction modes. R is the total number of bits for coding information of each block by using the entropy coding technique. Equation (2.32) is used to measure the overall rate-distortion cost involving both Luma and Chroma components.

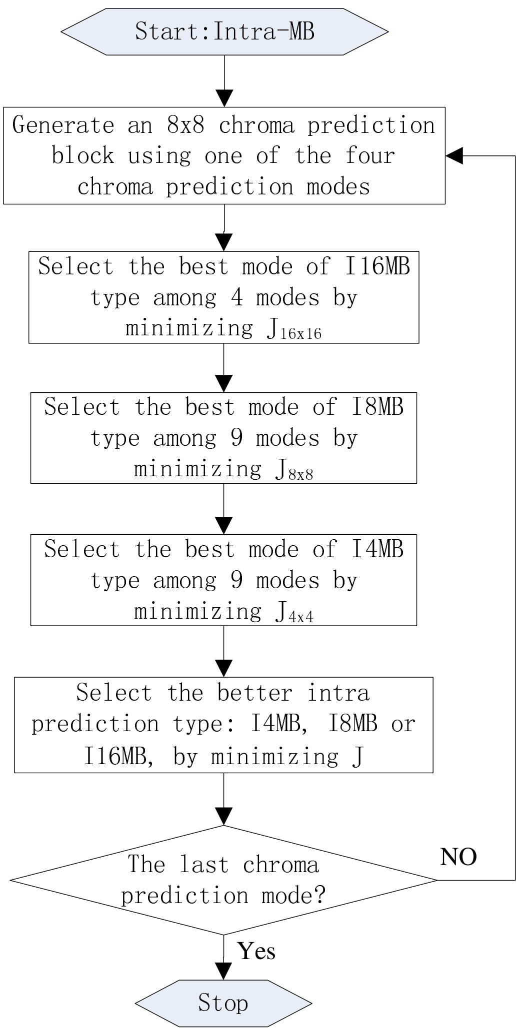


Figure 2.4. Flowchart of intra mode selection in high profile of H.264/AVC

From above analysis, we can see that both Luma and Chroma components are combined to perform the RDO calculation for intra-MB in the JM reference software [5]. The number of RDO equations ((2.29)-(2.31)) that have to be computed for an

MB is NC_{8x8}*(NL_{16x16} + NL_{8x8}*4 + NL_{4x4}*16), which is equal to 4x(4+9x4+9x16), where NC_{8X8}, NL_{16x16}, NL_{8x8},and NL_{4x4} represent the number of possible modes for an 8x8 Chroma block, 16x16 Luma block, 8x8 Luma block and 4x4 Luma block, respectively. As a result, the intra mode decision is especially complex, hence it is always desirable to have a better mode decision approach to simplify the process.

2.3 Lossless intra coding technique in H.264/AVC

Lossless coding technique plays an important role in perfectly preserving valuable information using less storage space. Many approaches have been proposed and are mainly developed for image coding [8, 42, 43]. A good example in standards is the JPEG lossless coding [8]. It has many applications, such as medical images, digital archives, digital documentations and so on. In the past few years, some further improvements of lossless image coding studies have been done. One of the representatives is an edge-directed prediction method in [9], which enables the predictor to be adaptive from smooth regions to edge regions. Note that the complexity is usually high for lossless image coding. With the development of video coding standards, high definition videos are gradually entering into our daily life. The demand for higher fidelity of encoding videos is ever increasing. It is highly possible that either the whole picture or some regions of a picture in a video need to be represented without any loss of fidelity for some demanding applications. Consequently, it is necessary to integrate the lossless coding technique into video coding standards, such as the H.264/AVC coding standard [1-4]. The H.264/AVC standard was originally designed for this purpose of compression. The compression efficiency of video coders can usually be achieved based on lossy coding technique. However, it also supports lossless video coding for some requirements.

The lossless coding method used in H.264/AVC standard can be considered as the state-of-art representative which is applied to the video coding standard. In H.264/AVC, lossless coding can be done by an intra pulse-code-modulation (IPCM) MB type, in which the values of the samples are encoded directly without prediction, transformation or quantization [1-4]. It is clear that this mode is not efficient since it requires many bits to encode an MB without any loss of fidelity. The experimental results also show that this type is rarely selected for an encoder with higher compression ratio. However it can provide a minimum upper bound on the number of bits to represent an MB with full accuracy [4].

As the development of H.264/AVC standard, a more effective lossless coding method was designed in the new profiles called FRExt High profiles [4]. In the new profiles, a transform-bypass lossless coding method was proposed to improve the efficiency of the lossless coding. This method uses intra prediction and entropy coding for encoding sample values. Figure 2.5 shows the lossless coding process and the lossy coding process of an intra residual block/MB in the FRExt High profiles. Note that, in the lossy video coding, the residual block should be transformed and quantized, and then the quantized DCT coefficients are scanned and encoded by entropy coding. Different from it, the residual block is directly scanned and encoded by entropy coding without the transformation or quantization process in the lossless coding. As a result, the encoding complexity can be reduced substantially but more bits are required to encode the video sequences.

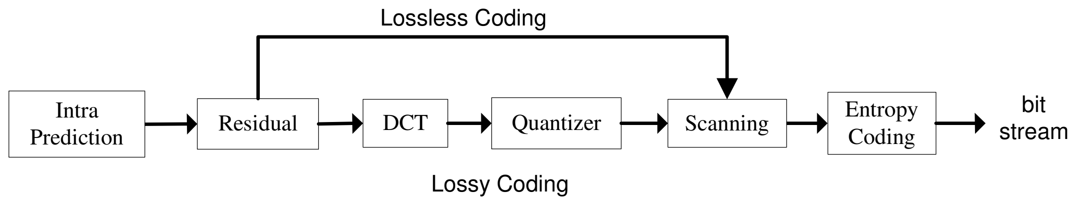


Figure 2.5. Lossless coding and lossy coding adopted in FRExt High profiles of H.264/AVC standard

From Figure 2.5, we can see that the intra prediction algorithm used to produce the residual is the same for both lossless coding and lossy coding. In FRExt High profiles, four basic types of intra prediction for Luma components are available: 16x16 Luma prediction (I16MB type), 8x8 Luma prediction (I8MB type) and 4x4 Luma prediction (I4MB type) and IPCM [1-4]. There are four prediction modes in the I16MB type and nine prediction modes in the I4MB/I8MB types, which are shown in Figures 2.2 and 2.3. Chroma components can be predicted using four modes, which are similar to those in I16MB type [1-4].

The new lossless coding method makes the encoder very efficient as compared with IPCM mode. However, we also note that the intra block/MB is predicted as a whole by an extrapolation of the neighboring reconstructed pixels due to a block-based coding structure in the H.264/AVC standard. As a result, the samples which are far from their references may not be predicted well due to poor correlation. Especially for the blocks predicted using the I8MB or I16MB type, the poor prediction is more obvious since a larger block size is used. Another drawback in the lossless coding of H.264/AVC framework is that the CAVLC entropy coding method is usually not efficient to encode the residual block directly since it is designed based on some characteristics of quantized DCT coefficients of the residual block.

Therefore, further improvement of lossless coding method in H.264/AVC standard is always desirable for the purpose of compression.

2.4 Unified intra coding technique in HEVC

As the next generation of video coding standard, HEVC, is now being designed to achieve significantly higher compression efficiency than all the existing video coding standards due to the adoption of many advanced coding tools. For example, up to 35 modes are available for intra prediction of each prediction unit in HEVC. This can provide more accurate predictions and thereby improve the compression efficiency of intra coding. In this thesis, we mainly focus on improving the intra coding in the HEVC standard.

2.4.1 Intra prediction in HEVC

Intra prediction is adopted to reduce the spatial redundancies in a frame. In the high profile of H.264/AVC, three block sizes (16x16, 8x8 and 4x4) with four, nine and nine modes as shown in Figures 2.2 and 2.3 are used adapting to different structures of a frame in intra prediction. Note that these three block sizes and the four or nine intra prediction modes are not flexible enough to represent complex structures. To provide more accurate predictions and to improve the coding efficiency of intra prediction, HEVC not only extends the set of PUs from 4x4 to 64x64 based on a tree structure, but also increases the set of intra prediction modes up to 35 by referring to the reconstructed samples from the left-down, left, up and up-right sides. In other words, when the LCU is up to 64x64, the size of PU can be 64x64, 32x32, 16x16, 8x8 and 4x4. For each PU, there are 35 intra prediction modes available in current version of the HEVC [4, 5]. Table 2.6 summarizes the 35 intra prediction modes adopted in HEVC [31-33]. It includes Planar mode, DC mode and

33 directional modes. Figure 2.6 shows the directions of the 33 directional modes in HEVC. They have the angles of +/-[0, 2, 5, 9, 13, 17, 21, 26, 32], respectively. To select the optimal mode, an encoder has to exhaust recursively all the combinations of CU and PU. For each PU, a three-stage approach is implemented to find the best intra prediction mode and the optimal transform structure in the test model HM of the HEVC standard. First, a rough mode decision (RMD) algorithm [34, 35] is used to select a subset of all intra prediction modes before the RDO process. Note that the three most probable modes as illustrated in Section 2.4.2 are always included in the subset. Second, the RD cost of each prediction mode in the subset is calculated without the use of residual quadtree (RQT) technique [36, 37]. The mode with the smallest RD cost is selected for coding the current PU. Third, the RQT technique is adopted to determine the optimal transform structure for the residual PU resulted from the difference between the current PU and the predicted PU corresponding to the best mode in the second step. As a result, the information left in the residual PUs is reduced significantly, and hence the coding efficiency of intra prediction is enhanced in the HEVC standard.

Table 2.6. Specification of intra prediction modes in HEVC

Intra prediction mode	**Associated names**
0	Planar
1	DC
2...34	Angular: {Hor+i| i=8,7,...,0}, {Hor-i| i=1,2,...,7}, {Ver-i| i=8,7,...,0}, {Ver+i| i=1,2,...,8}

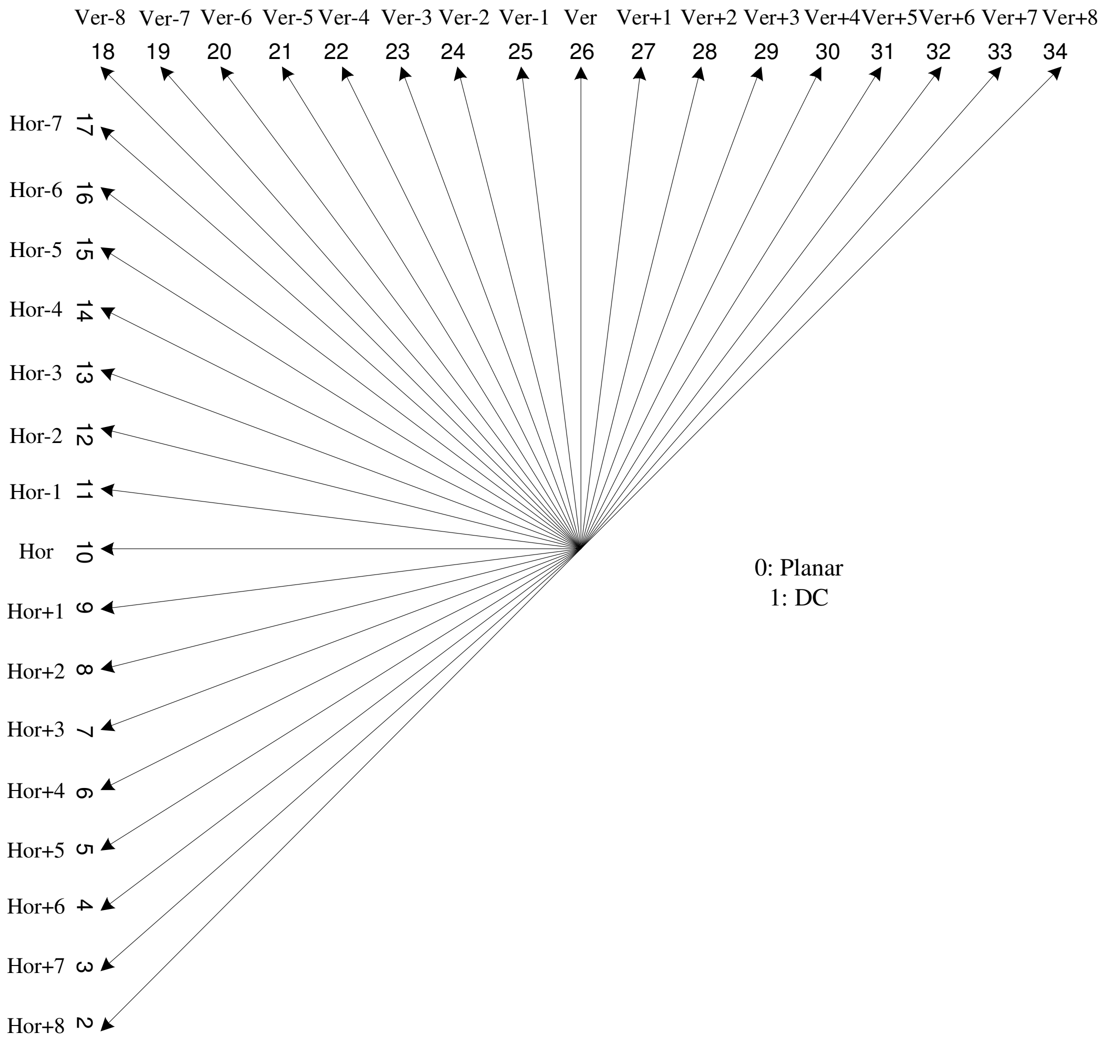


Figure 2.6. Up to 35 intra prediction modes in the current HEVC

2.4.2 Intra mode signaling in HM

After the optimal mode is selected for the current PU, the mode index should be included into the bitstream to inform the decoder. In Table 2.7, "prev_intra_luma_pred_flag" is a flag used to signal if the current mode (CurMode) is equal to the most probable mode (MPM). In the reference software JM of H.264/AVC, only one MPM with the smaller mode index between the two neighboring intra modes is used [5]. If the optimal mode is equal to MPM, "1" is encoded into the bitstream. If not, "0" is used to signal in the bitstream, and another symbol "rem_intra_luma_pred_mode" will be followed using a fixed length coding (FLC) method. This symbol is used to indicate which of the remaining intra prediction modes is selected. Since one out of nine modes should be signaled in the H.264/AVC, only 1 bit or 4 bits are enough for signaling each mode as shown in Table 2.7. However, one out of 35 modes in the current HM [35] is required to signal. As a result, more bits (2, 3 or 6 bits are required as shown in Table 2.7) should be assigned to signal the best mode for each PU in HM. Due to the use of more intra prediction modes, three MPMs (MPM0, MPM1 and MPM2) are set in the current HM7.0 [35]. The derivation of these three MPMs are shown in Table 2.8 according to HM7.0.

From Table 2.7, we can see that different codewords are set when the optimal mode is among the three MPMs. If none of the MPMs is equal to the optimal mode, six bits (0xxxxx) are required in the bitstream to indicate which mode is selected for the current PU. It is no doubt that the overhead bits required for signaling the mode information significantly increase the size of the bitstream due to the introduction of more intra prediction modes.

Table 2.7. Codewords for intra Luma prediction modes in H.264/AVC and HEVC

Standard	prev_intra_luma_pred_flag	Codewords
H.264/AVC (JM 18.0)	1	1
	0	0xxx (Note: xxx is used for binarizing **rem_intra_luma_pred_mode**, which is from 000 to 111)
HEVC (HM 7.0)	1	If CurMode=MPM0, Codewords=10; If CurMode=MPM1, Codewords=110; If CurMode=MPM2, Codewords=111.
	0	0xxxxx (Note: xxxxx is used for binarizing **rem_intra_luma_pred_mode**, which is from 00000 to 11111)

Table 2.8. Derivation of three MPMs in HM7.0

If left/top are unavailable or not intra, both left and up modes (LeftMode and UpMode) are set to DC.
o If LeftMode is the same as UpMode
  • If LeftMode is Planar or DC mode
    ▪ MPM0 = Planar
    ▪ MPM1 = DC
    ▪ MPM2 = Ver
  • else if LeftMode is not Planar or DC mode
    ▪ MPM0 = LeftMode
    ▪ MPM1 = ((LeftMode+29)%32)+2
    ▪ MPM2 = ((LeftMode-1)%32)+2
o else if LeftMode is not the same as UpMode
    ▪ MPM0 = LeftMode
    ▪ MPM1 = UpMode
  • If neither LeftMode nor UpMode is Planar
    ▪ MPM2 = Planar
  • else if (LeftMode+UpMode)<2
    ▪ MPM2 = Ver
  • else
    ▪ MPM2 = DC

2.4.3 Statistics of overhead bits for signaling intra mode in HM

Table 2.9 lists the percentages of bit consumption of the mode information for five different classes of sequences and six QP values using both All Intra Main (AIMain) and All Intra HE10 (AIHE10) configurations defined in [44]. From this table, we can see that the intra mode information accounts for more than 9% of the bitstream on average. It is really worth researching to reduce the bits used for signaling intra mode information.

Table 2.9. Bits consumption of signaling intra mode information in bitstream based on HM 7.0

	All Intra Main [%]						All Intra HE10 [%]					
QP	22	27	32	37	42	47	22	27	32	37	42	47
Class A	7.94	9.87	12.31	14.12	14.86	14.21	7.94	9.83	12.07	13.60	14.07	11.72
Class B	4.63	9.07	12.00	9.42	7.61	7.07	4.65	9.03	12.15	9.52	7.72	7.26
Class C	9.45	12.24	9.95	8.37	8.46	7.67	9.46	12.25	10.05	8.51	7.86	8.94
Class D	9.21	12.19	10.05	8.82	8.72	14.78	9.18	12.84	10.14	9.13	9.14	13.96
Class E	11.42	14.33	8.59	7.58	6.89	6.67	11.27	14.14	8.72	7.72	7.15	7.05
Average	8.53	11.54	10.58	9.66	9.31	10.08	8.50	11.62	10.63	9.70	9.19	9.79

Chapter 3 Early Detection Algorithms of ZQDCT Coefficients in H.264/AVC

3.1 Introduction

The objective of this chapter is to detect the ZQDCT coefficients before performing DCT, Q, IQ and IDCT processes such that the coding complexity of an encoder based on transformation and quantization techniques can be reduced.

Let us recall that Moon and Kim [6] proposed a sufficient condition to detect ZQDCT coefficients for the 4x4 integer DCT used in H.264/AVC [1-4] without leading to any RD performance degradation. This method can save 11%-36% computational saving without video-quality degradation based on the experimental results in [6]. However, some researchers found that a further saving could be made if a slight loss in the video quality is allowed. It is meaningful for real-time applications. As a consequence, the lossy detection method was developed. It is obvious that the lossy detection method is to detect as early as possible ZQDCT coefficients at the expense of a slight degradation of video quality. The related contributions can be seen in papers [16-21].

In order to further reduce the encoding time spending on DCT, Q, IQ and IDCT processes, we present two efficient detection algorithms for ZQDCT coefficients for video coding in this Chapter.

Compared with conventional detection methods of zero-quantized DCT coefficients used in H.264/AVC, the first proposed algorithm has two major features.

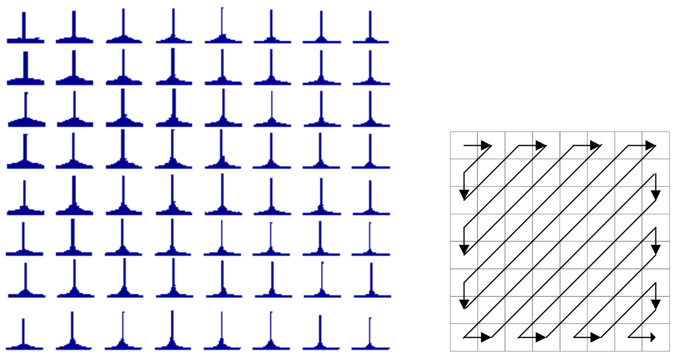
First, a new classification of patterns for DCT, Q, IQ and IDCT processes is proposed. By taking a zigzag scanning order into the classification, the quantized DCT coefficients can be coded efficiently. Second, the thresholds for detecting zero-quantized DCT coefficients are determined by combing the Gaussian distribution with a theoretical analysis of the DCT and quantization in H.264/AVC. Experimental results show that the proposed algorithm achieves an average timesaving of more than 40% compared with the algorithm in the reference software JM12.2 of H.264/AVC. When compared with other algorithms in the literature, it also gives the best performance in terms of both rate-distortion and time saving.

Note that the new classification of patterns for DCT, Q, IQ and IDCT processes shows better rate distortion performance in the first algorithm. By making use of the new patterns for DCT, Q, IQ and IDCT processes, another algorithm which can give different thresholds for the detection of ZQDCT coefficients in H.264/AVC is proposed. In this algorithm, a joint probability density function between each DCT coefficient and the corresponding SAD is formulated based on the Laplace distribution of motion-compensated frame difference. By forming a Lagrangian cost function between the false acceptance ratio (FAR) and false rejection ratio (FRR), this algorithm can select the optimal threshold for each DCT coefficient. Experimental results show that the Lagrangian FAR-FRR optimization algorithm matches the theoretical derivation, and is efficient for realization.

3.2 Early detection algorithm based on new patterns in H.264/AVC

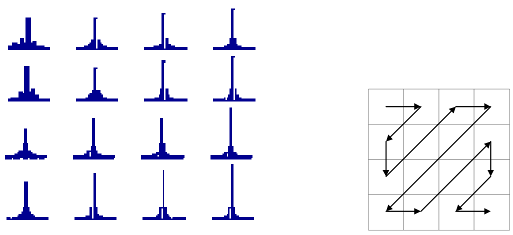
3.2.1 Patterns for DCT, Q, IQ and IDCT

The distributions of DCT coefficients in residual frames are firstly investigated. Figures 3.1 (a) and 3.2 (a) show an example of the histograms of 8x8 DCT coefficients and 4x4 DCT coefficients, respectively. From these two figures, several characteristics can be observed for both 8x8 DCT and 4x4 DCT. (i) In the same diagonal direction (from left-down to right-up), the DCT coefficients almost have the same distribution. In other words, the variances of the DCT coefficients are almost the same. (ii) From low frequency to high frequency, the variance of the distribution of each DCT coefficient is gradually decreased. Or we can say that, the variance of each DCT coefficient decreases gradually along 8x8/4x4 zigzag scanning orders, which are shown in Figure 3.1(b) and Figure 3.2(b), respectively.



(a) (b)

Figure 3.1. (a) Histogram of 8x8 DCT coefficients of the 1st residual frame for "Children.CIF", and (b) Zigzag scanning order of 8x8 DCT coefficients



(a) (b)

Figure 3.2. (a) Histogram of 4x4 DCT coefficients of the 1st residual frame for "Children.CIF", and (b) Zigzag scanning order of 4x4 DCT coefficients

Referring to [17, 18, 45-47], the variance $\sigma_Y^2(u,v)$ of the (u, v) DCT coefficient Y(u, v) can be denoted as

$$\sigma_Y^2(u,v) = \sigma_f^2 \left[ARA^T\right]_{u,u} \left[ARA^T\right]_{v,v} \cdot E_f(u,v) \cdot E_f(u,v), \tag{3.1}$$

where σ_f^2 denotes the variance of an input 4x4 residual block for DCT operation. Let ρ be the one-step correlation coefficient, in the horizontal or vertical direction [17, 18, 45], of R which is the correlation matrix. ρ can be calculated by the covariance r(·) of an input signal as $\rho = r(1,0)/\sigma_f^2$ and $\rho = r(0,1)/\sigma_f^2$ for horizontal and vertical directions, respectively [46]. From the definition of ρ, we can see that ρ is a local parameter which represents the characteristics of an input video sequence. For different kinds of input video sequences, we can obtain different ρ values. The threshold obtained by adjusting the value of ρ just means that this detection method is adjusted to represent better one kind of video sequences with the values of ρ equal to the adjusted ρ. After further simulation, we have found that the proposed algorithm can give a good performance in terms of rate distortion cost and time saving, when ρ is set to 0.4 for the tested video sequences as shown in Tables 3.3 and 3.4. For ρ with the value of 0.4, the variance of each 4x4 DCT coefficient can then be derived as shown in (3.2).

$$\sigma_Y = \sigma_f \begin{bmatrix} 1.7920 & 1.3788 & 1.0974 & 0.9228 \\ 1.3788 & 1.0608 & 0.8443 & 0.7100 \\ 1.0974 & 0.8443 & 0.6720 & 0.5651 \\ 0.9228 & 0.7100 & 0.5651 & 0.4752 \end{bmatrix} \tag{3.2}$$

Equation (3.2) shows that the variance of each DCT coefficient is proportional to the variance of pixels in the 4x4 residual block for frame coding. Along the zigzag scanning order, the variance decreases gradually. This theoretically confirms the observed characteristics abovementioned. Based on the observed characteristics, we propose a new scheme as shown in Figure 3.3 to classify the patterns for DCT, Q, IQ and IDCT to efficiently detect ZQDCT coefficients. As shown in Figure 3.3, the patterns are classified according to the zigzag scanning order in Figure 3.2(b). This classification makes more zeros concentrate in the high frequency not distributing in the middle part compared with the method in [17]. As a result, the scanned QDCT coefficients can be coded efficiently.

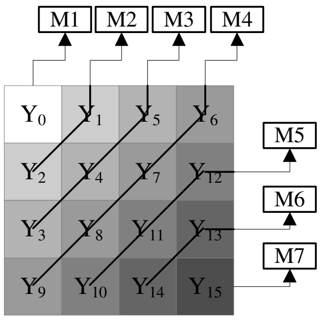


Figure 3.3. Eight modes from M0 to M7 for DCT, Q, IQ and IDCT

(M0: not including DCT coefficients; M1 including Y0; M2 including Y0, Y1 and Y2; ...; M7 including all 16 DCT coefficients)

Totally, eight modes as shown in Figure 3.3 are available according to the number of diagonal lines in a 4x4 block. However, we note that most of the DCT coefficients with high frequencies are usually quantized to zero for the purpose of efficient compression. Moreover, more details of the classification may require more time to determine which mode should be used, while the detailed classification contributes very little to the computational saving. As a result, we adopt four thresholds from T(0), T(1), T(2) and T(3) to detect five patterns of DCT, Q, IQ and IDCT processes as shown in Table 3.1.

Table 3.1. Proposed Relationship between SAD and pattern for DCT, Q, IQ and IDCT

Mode	**Condition**	**Pattern for DCT, Q, IQ and IDCT**
M0	$SAD < T(0)$	None
M1	$T(0) \le SAD < T(1)$	Y(0,0) Y(1,0) Y(0,1)
M2	$T(1) \le SAD < T(2)$	Y(0,0) Y(1,0) Y(2,0) Y(0,1) Y(1,1) Y(0,2)
M3	$T(2) \le SAD < T(3)$	Y(0,0) Y(1,0) Y(2,0) Y(3,0) Y(0,1) Y(1,1) Y(2,1) Y(0,2) Y(1,2) Y(0,3)
M4	$T(3) \le SAD$	Y(0,0) Y(1,0) Y(2,0) Y(3,0) Y(0,1) Y(1,1) Y(2,1) Y(3,1) Y(0,2) Y(1,2) Y(2,2) Y(3,2) Y(0,3) Y(1,3) Y(2,3) Y(3,3)

Note that the above analysis mainly gives focus on the detection of ZQDCT coefficient in frame MBs. For the detection of ZQDCT coefficient in field MBs, two major modifications may be required. First, it requires to set the one step correlation coefficient ρ in matrix R of (3.1). Note that ρ represents the one-step correlation coefficient in the vertical or horizontal direction. Usually, the vertical-direction and the horizontal-direction correlation coefficients are set to the same value for ZQDCT coefficient detection in frame MBs, such as in [17, 18]. However, they should be different in field MBs since the coded MBs involve interlacing. Second, we have to modify the classification of the patterns for DCT, Q, IQ and IDCT. If field MBs are used, the field scan in [5] is used for efficient compression based on H.264/AVC. In such case, the patterns for DCT, Q, IQ and IDCT should be changed corresponding to the field scan for efficient detection and compression. In other words, the patterns for DCT, Q, IQ and IDCT should be specified according to the scan mode for efficient compression.

3.2.2 Determination of thresholds

Since DCT coefficients can be modeled as a Gaussian distribution [48], the DCT coefficient Y(u, v) can be quantized to zero with a probability of 99.73% as given in [17] when (3.3) is satisfied.

$$3\sigma_Y(u,v) < (1-f)\cdot Qstep \tag{3.3}$$

Observing from [6] and [17], we note that σ_f can be approximated as a linear function of SAD. Hence, we assume that $\sigma_f = k \cdot SAD$. Substitute it into (3.2), we can have $\sigma_Y(u,v)$. Substitute $\sigma_Y(u,v)$ into (3.3), we have

$$SAD < \frac{(1-f)\cdot Qstep}{3k\cdot E_f(u,v)\cdot\sqrt{\left[ARA^T\right]_{u,u}\left[ARA^T\right]_{v,v}}} \tag{3.4}$$

Therefore, the threshold of our algorithm is

$$T\_WS(u,v) = \frac{(1-f)\cdot Qstep}{3k\cdot E_f(u,v)\cdot\sqrt{\left[ARA^T\right]_{u,u}\left[ARA^T\right]_{v,v}}} \tag{3.5}$$

From Table 2.4, we can note that when SAD is smaller than 3.3333*Qstep, the DC coefficient in (2.1) is not required to be computed theoretically in [6]. This condition is used to compute k in (3.5) by letting T_WS(0,0)= 3.3333*Qstep. As a result, k can be represented as

$$k = \frac{1}{3\sqrt{\left[ARA^T\right]_{0,0}\left[ARA^T\right]_{0,0}}} \tag{3.6}$$

By substituting (3.6) into (3.5), we obtain

$$T\_WS(u,v) = \frac{(1-f)\cdot Qstep \cdot \sqrt{\left[ARA^T\right]_{0,0}\left[ARA^T\right]_{0,0}}}{E_f(u,v)\cdot \sqrt{\left[ARA^T\right]_{u,u}\left[ARA^T\right]_{v,v}}} \tag{3.7}$$

As a result, four thresholds T_WS(0,0), T_WS(1,0), T_WS(2,0) and T_WS(3,0) corresponding to the thresholds from T(0) to T(3) in Table 3.1 are adopted to detect five patterns of DCT, Q, IQ and IDCT processes.

3.2.3 Experimental results

The proposed algorithm has been implemented in the reference software JM 12.2 [5] of H.264/AVC for the performance evaluation in terms of rate-distortion and encoding time. The test was based on the high profile of H.264/AVC standard. The prediction structure was IPPP.... A number of video sequences were used for this work as shown in Table 3.2. A total of 150 frames were encoded for each sequence. The QP values were set at 22, 27, 32 and 37 for intra frame coding, while 23, 28, 33 and 38 were used for inter frame.

Since the DCT and quantization processes take up about 16% of the total computations in a digital signal processor (DSP) based on H.264/AVC standard [49], it is always desirable to reduce the computation. It is common that quite a number of DCT coefficients are quantized to zero, especially for low bitrate coding. In such a case, it can save some computation if we directly compare the SAD which is obtained in the process of motion estimation with our determined threshold. In order to evaluate the efficiency of the proposed algorithm in terms of the percentage of computation saved during the implementation of the integer DCT, quantization, inverse quantization, and inverse DCT processes. The numbers of addition (ADD), subtraction (SUB), multiplication (MUL) and shift (SFT) operations are considered. Table 3.2 lists the saving of these four operations when QP is 28. Compared with the algorithms in [6] and [17], the proposed algorithm can save much computation in terms of the four operations for all tested video sequences as shown in Table 3.2. The saving in computation of the software encoder implies a reduction in power consumption of the hardware encoder, which is also in line with the concept of green computation advocated recently.

Table 3.2. Reduction percent of required computations in different algorithms

Seq.	**Operation**	**Previous Algorithm [6]** [%]	**Previous Algorithm [17]** [%]	**The proposed algorithm** [%]
Foreman (QCIF)	ADD/SUB	37.24	59.84	64.66
	MUL	31.53	56.04	69.29
	SFT	29.58	60.47	68.87
Akiyo (QCIF)	ADD/SUB	69.82	83.45	86.11
	MUL	65.38	81.47	88.46
	SFT	64.02	82.76	88.26
Paris (CIF)	ADD/SUB	37.57	59.36	64.36
	MUL	32.40	55.46	69.18
	SFT	30.59	60.02	68.75
Hall (CIF)	ADD/SUB	63.35	80.99	83.86
	MUL	56.67	78.88	86.30
	SFT	54.78	81.28	86.10
Crowdrun (720p)	ADD/SUB	21.54	29.63	39.69
	MUL	18.52	18.63	43.90
	SFT	17.45	27.49	43.45
Intotree (720p)	ADD/SUB	23.40	42.07	63.87
	MUL	16.68	25.19	70.53
	SFT	14.02	39.08	69.99

Besides the analysis of computational complexity, Table 3.3 gives the timing results on realizing various algorithms. The system for evaluating the algorithms was an Intel i7 950 system with 12GB of RAM. In this table, the average time spent on DCT, Q, IQ and IDCT based on [5] is listed in the column "Time (ms)" for the four QP values (23, 28, 33 and 38). On average, the above four processes usually take up about 4% of the whole encoding time based on our statistics on the JM12.2 platform in [5]. The percentages of time saving (ΔTIME) based on different detection methods for the above four processes are also shown in Table 3.3. From this table, we can see that our algorithm can save more time when compared with [6] or [17].

Table 3.3. Evaluation results about average time saving

Seq.		Time (ms) [5]	Δ TIME [%]		
			[6]	**[17]**	**Proposed Algorithm**
QCIF	Akiyo	1454	-48.71	-46.08	-64.82
	Monitor	1470	-18.85	-34.92	-49.78
	Carphone	1487	-40.68	-47.61	-53.12
	TableTennis	1485	-21.64	-32.06	-35.71
	Salesman	1623	-40.10	-47.54	-56.35
Average		**1504**	**-34.00**	**-41.64**	**-51.96**
CIF	Hall	5982	-36.60	-41.39	-54.06
	Paris	6416	-28.35	-40.63	-48.75
	Akiyo	5661	-54.50	-53.53	-58.44
	Foreman	6035	-38.36	-39.71	-50.60
	Mobile	6956	-17.58	-16.44	-28.80
Average		**6210**	**-35.08**	**-38.34**	**-48.13**
HDTV (720p)	CrowdRun	58591	-18.88	-20.28	-27.98
	InToTree	52915	-24.26	-28.16	-40.60
	ShuttleStart	46788	-59.68	-55.14	-62.55
Average		**52765**	**-34.27**	**-34.53**	**-43.71**

Table 3.4 summarizes the experimental results achieved by the proposed and the previous detection algorithms [6, 17] in terms of ΔPSNR and ΔBR, which are calculated based on the method in [50]. Since the algorithm in [6] is a lossless detection method, there is no loss in terms of rate distortion performance. However, the performance in time saving is limited as shown in Table 3.4. The algorithm in [17] can save more time compared with the method in [6] by a slight degradation in rate distortion performance. By contrast, our algorithm outperforms the algorithms in [6, 17] in terms of three measurements as shown in Tables 3.3 and 3.4. From these two tables, we can see that the proposed algorithm save more than 40% encoding time on average spending on DCT, Q, IQ and IDCT processes, while the rate distortion

performance of our algorithm is almost the same as that in [5]. Sometimes, it is even better than that in [5], such as the sequence of Monitor. Note that the PSNR increases 0.012dB with a decrease by 0.345% in bitrate. This is due to the fact that our algorithm adopts an efficient classification of patterns for DCT, Q, IQ and IDCT processes. This classification makes the quantized DCT coefficients be coded efficiently when compared with the method in [17]. On the other hand, the determination of thresholds for detecting ZQDCT coefficients plays an important role in the time saving of DCT, Q, IQ and IDCT processes. In a practical application, the proposed algorithm can be used to combine with any fast motion estimation or inter mode decision algorithm, such as those in [51-54], which allows achieving a faster and quality video coding with the IPPP structure, say for example.

Table 3.4. Evaluation results in terms of average rate-distortion performance

Seq.		**Previous Algorithm [6]**		**Previous Algorithm [17]**		**The proposed algorithm**	
		Δ PSNR [dB]	Δ BR [%]	Δ PSNR [dB]	Δ BR [%]	Δ PSNR [dB]	Δ BR [%]
QCIF	Akiyo	No change		-0.022	0.420	-0.019	0.359
	Monitor			0.005	-0.145	0.012	-0.342
	Carphone			-0.006	0.116	0.017	-0.348
	TableTennis			0.014	-0.305	0.012	-0.259
	Salesman			0.010	-0.225	0.009	-0.205
Average		**0**	**0**	**0.000**	**-0.028**	**0.006**	**-0.159**
CIF	Hall	No change		-0.012	0.212	-0.011	0.217
	Paris			-0.001	0.039	0.024	-0.431
	Akiyo			0.006	-0.183	0.012	-0.321
	Foreman			-0.007	0.161	0.001	-0.019
	Mobile			0.000	0.013	0.001	-0.011
Average		**0**	**0**	**-0.003**	**0.048**	**0.005**	**-0.113**
HDTV (720p)	CrowdRun	No change		0.000	0.002	0.001	-0.006
	InToTree			-0.001	0.033	-0.002	0.075
	ShuttleStart			0.001	-0.047	0.003	-0.099
Average		**0**	**0**	**0**	**-0.004**	**0.001**	**-0.010**

3.3 Early detection algorithm based on Lagrangian optimization of FAR-FRR in H.264/AVC

To obtain a better algorithm for ZQDCT coefficient detection based on normalized SAD, one must have a thorough understanding of the relationship between the DCT coefficients of a 4x4 residual block and its corresponding normalized SAD. In the following section, a joint probability density function (p.d.f.) for each DCT coefficient and the normalized SAD is firstly formulated based on the Laplace distribution model of the residual pixels, and then a Lagrangian FAR-FRR optimization (FFO) detection algorithm based on the joint pdf is proposed. Finally, a detailed analysis about the proposed algorithm is presented.

3.3.1 Formulation of mathematical model

To formulate the mathematical model for detecting the ZQDCT coefficients, the key problem is to find a better fitted and a tractable model for the distribution of the value of the residual pixel, x, produced by the block-based motion compensation technique in H.264/AVC [1-4]. In the literature, there are two distributions, Laplace distribution and Gaussian distribution, which are often adopted. Since the Gaussian distribution is expressed in terms of the square of x while the Laplace distribution is expressed in terms of absolute x, the Laplace distribution shows a tractable property during the formulation of our model. Moreover, we have investigated the distribution of x. Figure 3.4 shows an example about the distribution of the inter residual pixels corresponding to six QP values. The probability density function of the Laplace distribution is also shown in Figure 3.4. The experimental results show that the residual pixel, x, can be well approximated as a Laplace distribution model with zero mean and parameter λ_0 as (3.8). Therefore, the Laplace distribution is chosen as the base distribution of the residual pixel x.

$$f_X(x) = (\lambda_0/2)\exp(-\lambda_0|x|), \quad -\infty < x < \infty, \tag{3.8}$$

where λ_0 can be related to the variance σ_X^2 as follows

$$\sigma_X^2 = 2/\lambda_0^2 \tag{3.9}$$

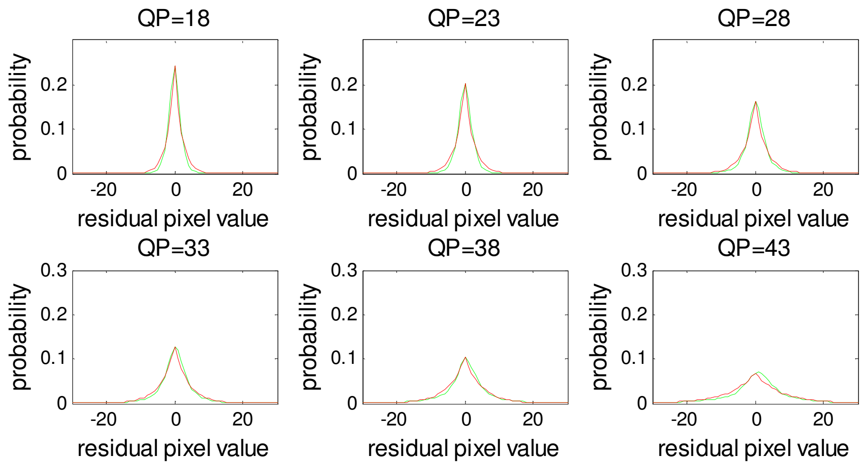


Figure 3.4. Distribution of the inter residual data for different Qstep values of Monitor.QCIF sequence (green: residual, red: Laplace)

Let M=|X|, and then the random variable M follows an exponential distribution (See Appendix A.1), labeled as M~exp($1/\lambda_0$). The p.d.f. of M can be expressed by

$$f_M(m) = \lambda_0 \exp(-\lambda_0 m), \qquad m \geq 0 \tag{3.10}$$

Assume that the absolute residual pixel value M is an independent random variable. Since the SAD of one 4x4 residual block is equal to

$$S = SAD = \sum_{i=0}^{15}|X_i| = \sum_{i=0}^{15} M_i, \tag{3.11}$$

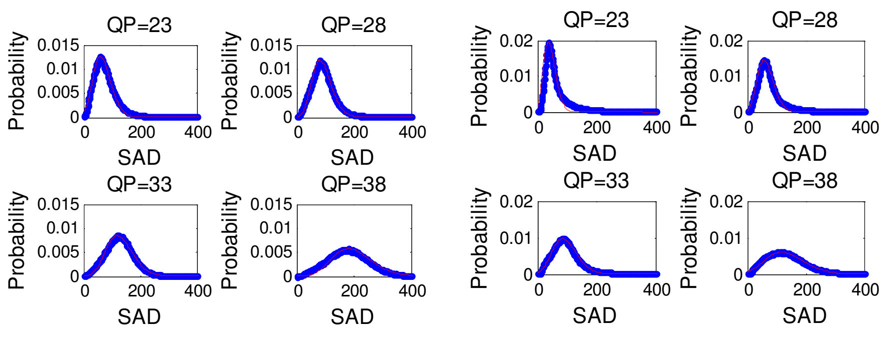
where S can be considered as the summation of independent identically distributed (IID) random variables M_i. Therefore, the p.d.f. of S can be written as follows (refer to Appendix A.2)

$$f_S(s)=\begin{cases}\dfrac{\lambda_0^{16}s^{15}}{(16-1)!}e^{-\lambda_0 s}, & s\geq 0\\ 0, & s<0\end{cases} \tag{3.12}$$

Note that S follows a Gamma distribution with parameters α=16 and β=1/λ_0, which can be labeled as S~Γ(α,β). For the Gamma distribution, the maximal $f_S(s)$ can be obtained when $s_{\text{model}}=15/\lambda_0$. However, there is a slight difference between the statistical distribution of SAD from the actual video sequences and the theoretical model. Let s_{actual} represent the position where the statistical model of SAD obtains the maximal value. Hence, the distribution of SAD can be considered as a shifted version of (3.12) along the direction of s by the size of s_0 as follows:

$$f_S(s)=\begin{cases}\dfrac{\lambda_0^{16}(s-s_0)^{15}}{(16-1)!}e^{-\lambda_0 (s-s_0)}, & s\geq s_0\\ 0, & s<s_0\end{cases}, \tag{3.13}$$

where $s_0=s_{\text{actual}}-s_{\text{model}}$. Figure 3.5 shows the Gamma model based on the theoretical analysis and the statistical model from two sequences. Each sequence includes 150 frames. Four QP values (23, 28, 33, 38) were used. These figures verify that the distribution of SAD can be well modeled by a shifted Gamma distribution.



(a)Mobile.QCIF (b)Bus.CIF

Figure 3.5. Theoretical model and statistical distribution of SAD for two sequences (blue: Histogram of SAD, red: Gamma Model)

Substituting (2.8) and (2.10) into (2.11), the absolute quantized-DCT (QDCT) coefficient is then given by (3.14).

$$\begin{aligned}|Z_{uv}| &= \left(|Y_{uv}| + f \cdot Qstep\right)/Qstep \\ &= g_{uv}/Qstep\end{aligned}, \tag{3.14}$$

where Y_{uv} is the DCT coefficient corresponding to the position (u,v) in Y of (2.1). From (3.14), we have

$$g_{uv} = |Y_{uv}| + f \cdot Qstep \tag{3.15}$$

Figure 3.6 shows that the distributions of (s, g_{uv}) corresponding to 16 (u,v) components.

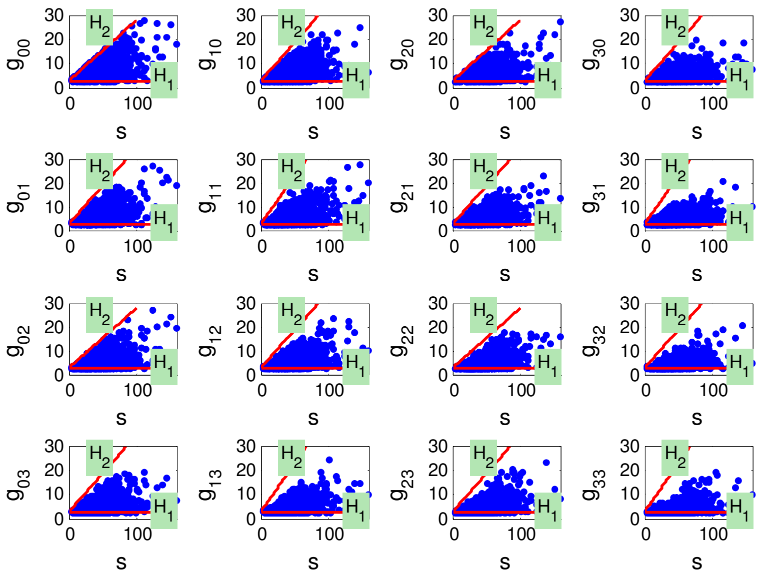


Figure 3.6. Boundaries of 16 (s, g_{uv}) distributions for Foreman.QCIF sequence

From Figure 3.6, we can see that the distribution of (s, g_{uv}) is located in a region bounded by two lines H_1 and H_2.

$$H_1\colon\ g_{uv} = f \cdot Qstep \tag{3.16}$$

$$H_2\colon\ g_{uv} = K_{uv} \cdot s + f \cdot Qstep\ , \tag{3.17}$$

where K_{uv} denotes the slope of line H_2. From (2.2) and (2.15), we have

$$\begin{aligned} |W_{uv}| &= \left| \sum_{i=0}^{3} \sum_{j=0}^{3} x(i,j) \cdot A(i,u) \cdot A(j,v) \right|, \\ &\leq \max_{0 \leq \langle i,j \rangle \leq 3} \left(A(i,u) \cdot A(j,v) \right) \times SAD \end{aligned} \tag{3.18}$$

where $\max_{0 \leq \langle i,j \rangle \leq 3} \left(A(i,u) \cdot A(j,v) \right)$ for all (u,v) components can be given as follows based on (2.16),

$$[C_{uv}] = \left[\max_{0 \leq \langle i,j \rangle \leq 3} \left(A(i,u) \cdot A(j,v) \right) \right] = \begin{bmatrix} 1 & 2 & 1 & 2 \\ 2 & 4 & 2 & 4 \\ 1 & 2 & 1 & 2 \\ 2 & 4 & 2 & 4 \end{bmatrix} \tag{3.19}$$

We have

$$\begin{aligned} |W_{uv}|_{\max} &= \max_{0 \leq \langle i,j \rangle \leq 3} \left(A(i,u) \cdot A(j,v) \right) \cdot SAD \\ &= C_{uv} \cdot s \end{aligned} \tag{3.20}$$

Substituting (3.15) and (3.20) into (3.17), we have

$$\begin{aligned} K_{uv} &= |Y_{uv}|_{\max} / s = |W_{uv}|_{\max} \cdot E_{uv} / s \\ &= C_{uv} \cdot E_{uv} \end{aligned} \tag{3.21}$$

Note that DCT coefficient can be modeled as a Laplace distribution. This model has been used extensively in image and video processing [48, 55-57]. It is always a popular choice to balance out the simplicity of a model and fidelity to empirical data. In our research, we also consider that each DCT coefficient Y_{uv} has a Laplace distribution with parameter λ_{uv}, which can be calculated by the standard deviation σ_{uv} as $\lambda_{uv} = \sqrt{2} / \sigma_{uv}$,

$$f_{Y_{uv}}(y_{uv}) = (\lambda_{uv}/2) \exp\left(-\lambda_{uv} |y_{uv}|\right) \quad , \quad \text{for } -\infty < y_{uv} < \infty \tag{3.22}$$

Similar to (3.10), the absolute value of Y_{uv} has an exponential distribution, and the p.d.f. is

$$f_{|Y_{uv}|}(|y_{uv}|) = \lambda_{uv} \exp\left(-\lambda_{uv} |y_{uv}|\right) \tag{3.23}$$

Making use of (3.15) and (3.23), the p.d.f. of G_{uv} can be expressed as

$$f_{G_{uv}}(g_{uv}) = \begin{cases} \lambda_{uv}\, e^{-\lambda_{uv}(g_{uv} - f\cdot Qstep)}, & g_{uv} \ge f\cdot Qstep \\ 0, & otherwise \end{cases} \tag{3.24}$$

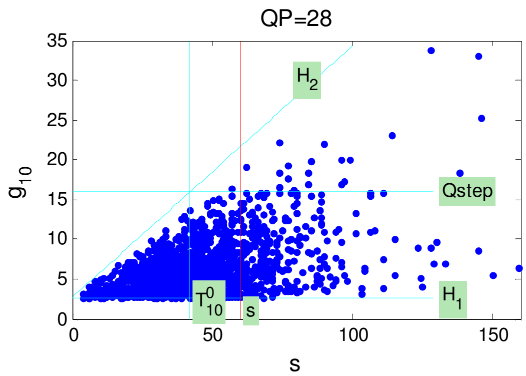


Figure 3.7. Example to illustrate the relationship between some parameters for Foreman.QCIF

Given S=s (the red line in Figure 3.7), the conditional pdf of variable G_{uv} can be approximated as an exponential distribution as shown in (3.25) based on our investigation. Similar to λ_{uv} used in (3.23), λ'_{uv} can be calculated by the standard deviation σ'_{uv} as $\lambda'_{uv} = \sqrt{2}/\sigma'_{uv}$. From Figure 3.7, variable G_{uv} is observed to lie in the interval [$f\cdot Qstep$, $K_{uv}\cdot s+f\cdot Qstep$] given S=s. Hence, the conditional pdf of the variable G_{uv} can be written as (3.25).

$$f_{G_{uv}|S}(g_{uv}|s) = \begin{cases} C\lambda'_{uv}\, e^{-\lambda'_{uv}(g_{uv} - f\cdot Qstep)}, & f\cdot Qstep \le g_{uv} \le K_{uv}\cdot s + f\cdot Qstep \\ & and\ 0 \le s \le 255\times 16 \\ 0, & otherwise \end{cases} \tag{3.25}$$

Note that the distribution of G_{uv} is equal to zero outside the interval [$f\cdot Qstep$, $K_{uv}\cdot s+f\cdot Qstep$]. Hence, parameter C can be determined by

$$\int_{f\cdot Qstep}^{K_{uv}\cdot s + f\cdot Qstep} f_{G_{uv}|S}(g_{uv}|s)\, dg_{uv} = 1 \tag{3.26}$$

Calculating from (3.26), we can obtain that $C = \left(1 - e^{-\lambda'_{uv} \cdot K_{uv} \cdot s}\right)^{-1}$. Note, however, that when s is smaller, the distribution approaches to H_2 with a value which is not zero as shown in Figure 3.7. In order to fit for such a case which is critical for the conditional model, (3.27) is used to approximately calculate the shape-controlled parameter σ'_{uv}.

$$3\sigma'_{uv} \approx K_{uv} \cdot s \tag{3.27}$$

Hence λ'_{uv} can be obtained as shown in (3.28) by substituting (3.27) into $\lambda'_{uv} = \sqrt{2}/\sigma'_{uv}$.

$$\lambda'_{uv} \approx 3\sqrt{2}/(K_{uv} \cdot s) \tag{3.28}$$

By substituting C and (3.28) into (3.25), it gives (3.29).

$$f_{G_{uv}|S}(g_{uv}|s) = \begin{cases} \dfrac{3\sqrt{2}}{\left(1-e^{-3\sqrt{2}}\right)K_{uv} \cdot s} \mathrm{e}^{-\frac{3\sqrt{2}}{K_{uv} \cdot s}(g_{uv} - f \cdot Qstep)}, & f \cdot Qstep \le g_{uv} \le K_{uv} \cdot s + f \cdot Qstep \\ & \mathit{and}\ 0 \le s \le 255 \times 16, \\ 0, & \mathit{otherwise} \end{cases} \tag{3.29}$$

Consequently, the joint probability density function $f_{SG}(s,g)$ of S and G_{uv} can be given by (3.30).

$$\begin{aligned} & f_{SG_{uv}}(s, g_{uv}) \\ & = f_S(s) \times f_{G_{uv}|S}(g_{uv}|s) \\ & = \begin{cases} \dfrac{\lambda_0^{16}(s-s_0)^{15}}{15!} \mathrm{e}^{-\lambda_0(s-s_0)} \times \dfrac{3\sqrt{2}}{\left(1-e^{-3\sqrt{2}}\right)K_{uv} \cdot s} \mathrm{e}^{-\frac{3\sqrt{2}}{K_{uv} \cdot s}(g_{uv} - f \cdot Qstep)}, & f \cdot Qstep \le g_{uv} \le K_{uv} \cdot s + f \cdot Qstep \\ & \mathit{and}\ 0 \le s \le 255 \times 16 \\ 0, & \mathit{otherwise} \end{cases} \end{aligned}$$

u,v=0,...,3 (3.30)

So far, the joint probability density functions between the DCT coefficients and their corresponding SAD are achieved. In the next section, a Lagrangian FAR-FRR optimization (FFO) detection method will be suggested in details based on the joint probability density functions.

3.3.2 Lagrangian FFO detection method of ZQDCT coefficients

Let us define J_1 to measure the ratio of the number of non-ZQDCT coefficients being falsely classified as ZQDCT coefficients to the total number of non-ZQDCT coefficients. J_2 is defined to represent the ratio of the number of ZQDCT coefficients being falsely classified as non-ZQDCT coefficients to the total number of ZQDCT coefficients. In fact, the two parameters were initially proposed to measure the quality of biometric recognition systems [58] with another two names: false acceptance ratio (FAR) and false rejection rate (FRR). Subsequently, these two parameters were also introduced to evaluate the detection capability of different algorithms in [58]. Referring to Figure 3.7, J_1 and J_2 can be expressed as follows:

$$J_1(T_{uv}) = \frac{1}{N_1}\int_{T_{uv}^0}^{T_{uv}}\int_{Qstep}^{K_{uv}\cdot s+f\cdot Qstep} f_{SG_{uv}}(s, g_{uv})dg_{uv}ds \tag{3.31}$$

$$J_2(T_{uv}) = \frac{1}{N_2}\int_{T_{uv}}^{T^{\max}}\int_{f\cdot Qstep}^{Qstep} f_{SG_{uv}}(s, g_{uv})dg_{uv}ds\,, \tag{3.32}$$

where N_1 represents the number of non-ZQDCT coefficients, and N_2 represents the number of ZQDCT coefficients. They can be calculated by (3.33) and (3.34), respectively. The closed form of N_1 and N_2 is too long to be written here. However, it can be easily calculated by some mathematical softwares.

$$N_1 = \int_{T_{uv}^0}^{T^{\max}}\int_{Qstep}^{K_{uv}\cdot s+f\cdot Qstep} f_{SG_{uv}}(s, g_{uv})dg_{uv}ds \tag{3.33}$$

$$N_2 = \int_{f\cdot Qstep}^{Qstep}\int_{\frac{g_{uv}-f\cdot Qstep}{K_{uv}}}^{T^{\max}} f_{SG_{uv}}(s, g_{uv})dsdg_{uv} \tag{3.34}$$

In (3.33), T_{uv}^0 denotes the lower limit of integral variable s as shown in Figure 3.7. In theory, T_{uv}^0 can be expressed as

$$T_{uv}^{0} = (1-f)\cdot Qstep/K_{uv} \tag{3.35}$$

Assume that n-bit grey level is adopted for an encoder. The maximum limit of SAD based on a 4x4 residual block is equal to $(2^n-1)\mathrm{x}4^2$. Thus, the upper limit of integral variable s, T^{max}, is given by (3.36).

$$T^{max} = (2^n-1)\mathrm{x}4^2 \tag{3.36}$$

Observed from (3.31) and Figure 3.7, we can see that a smaller J_1 makes a smaller number of non-ZQDCT coefficients be misclassified to zero, which means that the degradation of video-quality is less. Note that J_2 generally has a larger value when J_1 is smaller. Following a similar observation, (3.32) shows that if J_2 is smaller, a larger number of ZQDCT can be detected. As a result, much computational complexity can be reduced, but J_1 is larger. Therefore, we can see that, for efficient detection of ZQDCT coefficients, smaller J_1 and J_2 values are more desirable. From (3.31) and (3.32), it is clear that both J_1 and J_2 are the functions of T_{uv}. Hence, the tradeoff between J_1 and J_2 can be normally achieved by forming a Lagrangian cost function, which is written as follows:

$$J(T_{uv}) = J_1(T_{uv}) + \lambda J_2(T_{uv}),\ \text{for u,v=0,...3,} \tag{3.37}$$

where J is the Lagrangian cost, which is a function of T_{uv}. λ is the Lagrange multiplier, which represents a tradeoff between J_1 and J_2. The range of λ should be not less than zero theoretically.

To optimize both J_1 and J_2, the Lagrangian cost J should be minimized. Since the FAR-FRR curve is convex, and both J_1 and J_2 are differentiable, the minimal J can be given by

$$\frac{dJ(T_{uv})}{dT_{uv}} = \frac{dJ_1(T_{uv})}{dT_{uv}} + \lambda\frac{dJ_2(T_{uv})}{dT_{uv}} = 0 \tag{3.38}$$

After some mathematical manipulations (see Appendix A.3), T_{uv} can be obtained as follows:

$$T_{uv} = \frac{3\sqrt{2}(1-f)\cdot Qstep}{K_{uv}\times \ln\left[\left(1+\lambda\frac{N_1}{N_2}\right)\Big/\left(e^{-3\sqrt{2}}+\lambda\frac{N_1}{N_2}\right)\right]} \quad \text{u,v=0,...3} \tag{3.39}$$

Therefore, the threshold for the normalized SAD by Qstep can be given as (3.40).

$$TN_{uv} = \frac{3\sqrt{2}(1-f)}{K_{uv}\times \ln\left[\left(1+\lambda\frac{N_1}{N_2}\right)\Big/\left(e^{-3\sqrt{2}}+\lambda\frac{N_1}{N_2}\right)\right]} \tag{3.40}$$

3.3.3 Analysis of the Lagrangian FFO detection method

Equation (3.40) shows that TN_{uv} depends on the ratio of N_1 and N_2. From (3.35) and (3.36), the ratio γ of N_1 and N_2 can be obtained as follows:

$$\begin{aligned}\gamma &= N_1/N_2 = N_1/(N-N_1)\\ &= \frac{I_2\left(T_{uv}^0\right)-e^{-3\sqrt{2}}I_1\left(T_{uv}^0\right)}{\left(1-e^{-3\sqrt{2}}\right)I_1(0)-\left[I_2\left(T_{uv}^0\right)-e^{-3\sqrt{2}}I_1\left(T_{uv}^0\right)\right]}\end{aligned} \tag{3.41}$$

where

$$I_1(x) = \int_x^{T^{\max}} s^{15}e^{-\lambda_0\cdot s}ds \tag{3.42}$$

and

$$I_2(x) = \int_x^{T^{\max}} s^{15}e^{-\lambda_0\cdot s}e^{3\sqrt{2}(f-1)Qstep/(k_{uv}\cdot s)}ds \tag{3.43}$$

Equations (3.41)-(3.43) show that γ depends on λ_0, i.e., σ_X (in (3.9)) which represents the inherent property of an input sequence. As shown in Figure 3.4, parameter σ_X usually increases with an increase of *Qstep* for a given sequence. In other words, σ_X is also highly related to *Qstep*. The reason is that the Lagrange multiplier λ_{MOTION} in (3.44) becomes larger with an increase of QP (Qstep), so that the rate term, R(m-p), in (3.44) is increasingly important with the increase of λ_{MOTION}. As a result, a larger SAD will be chosen during the motion estimation process based on the RDO formula (3.44). In other words, the probability of smaller SAD becomes small so that σ_X tends to be large for a larger *Qstep*.

$$J(\mathbf{m},\lambda_{MOTION}) = SAD(s,c(\mathbf{m})) + \lambda_{MOTION} \cdot R(\mathbf{m}-\mathbf{p}) , \tag{3.44}$$

where J denotes the Lagrangian cost for the motion estimation process. $\lambda_{MOTION} = \sqrt{\lambda_{MODE}}$, and λ_{MODE} can be expressed as $\lambda_{\mathrm{MODE}} = 0.85 \times 2^{(QP-12)/3}$ if no B slices are used. *SAD*(*s*,*c*(*m*)) is the SAD between the original block *s* and the estimated block *c*(*m*) by motion estimation. $\mathbf{m} = (m_x, m_y)^T$ is the motion vector, and $\mathbf{p} = (p_x, p_y)^T$ is the prediction for the motion vector. *R*(*m*-*p*) denotes the number of bits used to encode the difference between ***m*** and ***p***. To sum up, σ_X should be adaptive to the input sequence and *Qstep*. However, it is slightly difficult to determine a value for versatile input sequences. Let us consider that the relationship between σ_X and *Qstep* is comparatively stable. *Qstep* is the only adaptive factor considered to ease the actual implementation while the adaptivity to the input sequence is sacrificed in the computation of σ_X. The relationship between *Qstep* and standard deviation σ_X has been investigated with extensive simulation work, and we have arrived at a linear approximation as shown in (3.45) to describe the relationship.

$$\sigma_X = a \cdot Qstep + b , \tag{3.45}$$

where a =0.3109, and b=10.22 which are used in the current realization, and they can give better simulation results.

Another parameter, s_{actual}, which is used to represent the position where the maximal probability of SAD happens, can be approximated as (3.46) in a similar way to (3.45).

$$s_{actual} = 1.632 \times Qstep + 53.53 \tag{3.46}$$

Note that λ in (3.37) is an important parameter, which provides a performance tradeoff between FAR and FRR. A big value of λ indicates that the importance of lower FRR is greatly emphasized in the FFO process. As a consequence, the resultant threshold (3.39) is larger which corresponds to a relatively lower FRR and a relatively higher FAR. Consequently, the encoder gives a larger distortion but a lower complexity. On the contrary, a smaller value of λ corresponds to a lower FAR with a higher FRR. This is necessary to have a better video quality. Minimizing the Lagrangian cost for a given λ is equivalent to finding a threshold TN_{uv}, where the optimal tradeoff between FAR and FRR can be found. This means that the optimal threshold TN_{uv} is actually controlled by λ.

Note that each λ can give a set of optimal thresholds. Given a larger λ, larger thresholds can be obtained, and vice versa. Along with the increase of thresholds, more ZQDCT coefficients can be detected, but much degradation in terms of the quality of the reconstructed videos may be resulted in. To keep a reasonable performance in terms of the quality of the reconstructed videos, λ is suggested to be within [0, 1/512] based on the simulation work in Section 3.3.5. Furthermore, a lookup table should be used to save computation in the actual implementation. It is significant to point out that not much extra computation is introduced during the fast encoding processes by the proposed algorithm. The most complex part in this algorithm is the computation of TN_{uv}, which can be done offline, before the detection algorithm of ZQDCT coefficients.

3.3.4 Refinement of T_{uv} (TN_{uv})

According to our observation in Figure 3.2, the proposed scheme in Table 3.1 of Section 3.2 is adopted to classify the patterns for DCT, Q, IQ and IDCT to efficiently simplify these processes. As shown in Table 3.1, the patterns are classified according to the zigzag scanning order in Figure 3.3. This classification makes more zeros concentrate in the high frequency not distributing in the middle part compared with the method in [12-19]. As a result, the scanned QDCT coefficients can be coded efficiently.

Thresholds from T(0) to T(3) can be calculated based on T_{00}-T_{30} in (3.39), respectively. Based on Table 3.1, we have T(0)<T(1)<T(2)<T(3). From (3.39), we can see that T_{uv} depends on K_{uv}, which is calculated by (3.21). To refine the threshold T_{uv}, the following limitations have been made.

$$\alpha_1 C_{\max}(1,0)\cdot E_f(1,0) < \alpha_0 C_{\max}(0,0)\cdot E_f(0,0) \tag{3.47}$$

$$\alpha_2 C_{\max}(2,0)\cdot E_f(2,0) < \alpha_1 C_{\max}(1,0)\cdot E_f(1,0) \tag{3.48}$$

$$\alpha_3 C_{\max}(3,0)\cdot E_f(3,0) < \alpha_2 C_{\max}(2,0)\cdot E_f(2,0) \tag{3.49}$$

Let α_0=1 for simplicity. From (3.47)-(3.49), we have

$$\alpha_1 < \sqrt{10}/4 \tag{3.50}$$

$$\alpha_2 < \alpha_1 2\sqrt{10}/5 \tag{3.51}$$

$$\alpha_3 < \alpha_2 \sqrt{10}/4 \tag{3.52}$$

In our experiments, we set $\alpha_1 = \sqrt{10}/4\times 90\%$, $\alpha_2 = \alpha_1 2\sqrt{10}/5\times 90\%$ and $\alpha_3 = \alpha_2\sqrt{10}/4\times 90\%$. In this condition, K_{uv} can be calculated as follows

$$K_{uv} = \alpha_u \cdot C_{\max}(u,v)\cdot E_f(u,v) \qquad \text{for u=0,1,...,3} \tag{3.53}$$

Substitute (3.53) into (3.39), the refined values of T_{00}-T_{30} can be calculated. Therefore, we can have T(0)-T(3) in Table 3.1 by letting them equal the refined values of T_{00}-T_{30}, respectively.

3.3.5 Experimental Results

To evaluate the performance of the proposed algorithm, the proposed algorithm has been implemented in the H.264/AVC reference software JM 12.2 [5] for the rate-distortion performance evaluation. The test was based on the high profile with specifications provided in [59]. The prediction structure was IPPP.... Three video sequences with QCIF format (Monitor, Carphone and Container), three with CIF format (Hall, Paris and Akiyo) and two with 720p format (Crowdrun and Intotree) were used for this work. A total of 150 frames were encoded for each sequence. Four QP values 22, 27, 32 and 37 were used for intra frame, while 23, 28, 33 and 38 were used for inter frame. The average changes of PSNR (ΔPSNR) and bitrate (ΔBR) were computed using the method in [50]. The averaged reduction of encoding time (ΔTIME) for those four QP values with respect to JM12.2 in [5] during DCT, Q, IQ and IDCT processes was also measured. In the following tables, positive values mean increase, and negative values mean decrease.

3.3.5.1 Experimental results about rate-distortion performance

Tables 3.5 and 3.6 show the experimental results in terms of ΔPSNR and ΔBitrate, respectively. Due to adopting a lossless detection algorithm, there is no change in terms of rate distortion performance for algorithm in [6] compared with H.264/AVC. However, the reduction of encoding complexity based on [6] is limited based on the experimental results in Table 3.7. From Table 3.5, we can see that there are 0.003dB decrease on average in terms of PSNR based on the method in [17]. By contrast, the better PSNR performance can be obtained when λ ranges from 0 to 1/32 based on our averaged results, such as the averaged 0.004dB increase of PSNR can be seen when λ is equal to 0, and the averaged 0.001dB increase of PSNR for λ with the value of 1/32. According to Table 3.6, we can see that the bitrates also vary for different λ values. In most cases, the bitrates decrease since some non-ZQDCT coefficients falsely classified as ZQDCT coefficients. On average, our algorithm can give the better performance (such as reduction from 0.124% to 0.020%) in bitrate than the method in [17] (0.048%) with λ between 0 and 1/32 as shown in Table 3.6.

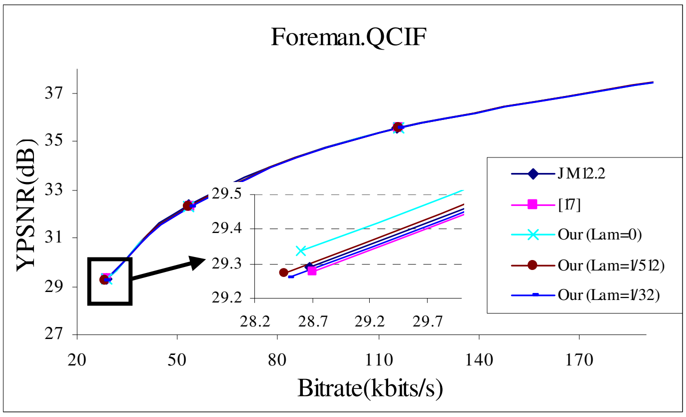
Figure 3.8 shows the RD curves for three sequences with different sizes. Note that the RD performance is similar to that of JM12.2 for most values of λ, and sometimes even much better than that in the H.264/AVC as shown in Figure 3.8, such as for λ with the value of 0 or 1/512.

Table 3.5. Change of PSNR (ΔPSNR) based on [50]

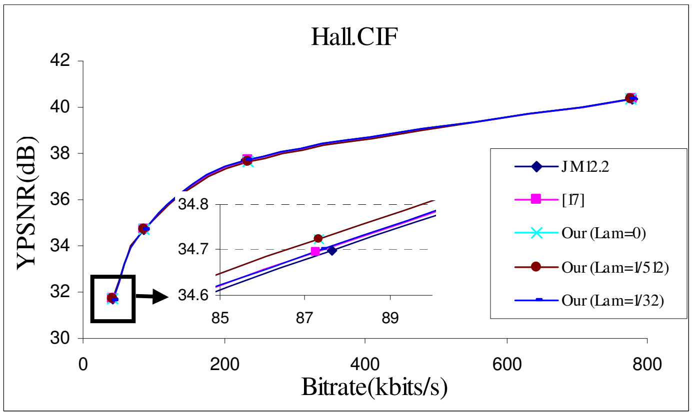
Sequence	Method in [17]	Our method						
		λ=0	λ=1/512	λ=1/256	λ=1/128	λ=1/64	λ=1/32	λ=1/16
Monitor	0.005	0.015	0.013	0.008	0.012	0.008	0.010	0.009
Carphone	-0.006	0.004	0.002	-0.007	0.001	0.020	0.000	-0.018
Container	-0.015	0.006	0.008	0.001	-0.011	-0.004	-0.006	-0.011
Hall	-0.012	-0.011	-0.011	-0.002	-0.002	-0.001	-0.008	-0.016
Paris	-0.001	0.002	0.006	0.017	0.020	0.006	0.009	-0.002
Akiyo	0.006	0.014	0.013	0.013	-0.005	-0.011	0.007	-0.020
Crowdrun	0.000	0.000	0.000	0.000	0.000	0.001	-0.002	-0.007
Intotree	-0.001	0.001	0.000	-0.003	-0.001	-0.001	-0.002	-0.006
Average	-0.003	0.004	0.004	0.003	0.002	0.002	0.001	-0.009

Table 3.6. Change of Bitrate (ΔBitrate) based on [50]

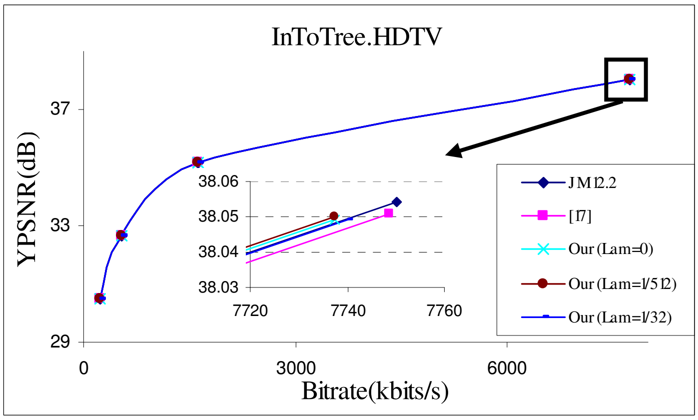
Sequence	Method in [17]	Our method						
		λ=0	λ=1/512	λ=1/256	λ=1/128	λ=1/64	λ=1/32	λ=1/16
Monitor	-0.145	-0.346	-0.296	-0.244	-0.280	-0.232	-0.226	-0.244
Carphone	0.116	-0.082	-0.045	0.148	-0.016	-0.415	-0.012	0.396
Container	0.313	-0.154	-0.198	-0.012	0.288	0.078	0.127	0.238
Hall	0.212	0.053	0.056	-0.007	-0.132	-0.199	0.137	0.281
Paris	0.039	-0.034	-0.110	-0.321	-0.353	-0.094	-0.133	0.021
Akiyo	-0.183	-0.379	-0.354	-0.346	0.099	0.236	-0.183	0.453
Crowdrun	0.002	0.002	0.008	0.004	0.004	-0.006	0.052	0.136
Intotree	0.033	-0.052	-0.036	0.111	0.021	-0.004	0.079	0.286
Average	0.048	-0.124	-0.122	-0.083	-0.046	-0.080	-0.020	0.196



(a)



(b)



(c)

Figure 3.8. Rate Distortion curves for three sequences with different sizes

3.3.5.2 Experimental results about encoding time

Tables 3.7 lists the averaged saving of time spending on DCT, Q, IQ, and IDCT for those four observed QP values achieved by the proposed and the previous detection algorithms [6, 17]. On average, the time can be saved from 45.613% to 47.772% when λ is from 0 to 1/32 as shown in Table 3.7. By contrast, the time savings from the algorithms in [6] and [17] are always less than the cases using our algorithm. As shown in Table 3.7, only 31.186% and 36.655% reduction can be observed averagely based on the methods in [6] and [17], respectively.

Table 3.7. Reduction of encoding time (ΔTime)

Seq.		**[6] (%)**	**[17] (%)**	**The proposed algorithm (%)**		
				λ=0	λ=1/512	λ=1/32
QCIF	Monitor	-18.849	-34.915	-44.750	-47.596	-48.861
	Carphone	-40.679	-47.614	-52.361	-52.582	-55.066
	Container	-27.372	-26.729	-50.692	-52.495	-54.348
CIF	Hall	-36.598	-41.393	-52.384	-52.727	-53.125
	Paris	-28.346	-40.629	-41.470	-41.988	-42.250
	Akiyo	-54.504	-53.527	-57.776	-58.215	-61.232
HDTV (720p)	CrowdRun	-18.876	-20.276	-26.050	-26.182	-27.027
	InToTree	-24.261	-28.156	-39.424	-39.518	-40.266
Average		**-31.186**	**-36.655**	**-45.613**	**-46.413**	**-47.772**

From all the results listed in Tables 3.5-3.7, we can see that our algorithm can provide more time savings in the DCT, Q, IQ and IDCT processes with a smaller loss in encoding efficiency than the algorithms in [6] and [17], when λ in (3.39) is set in the range [0,1/32]. Generally speaking, more time can be saved based on the proposed algorithm with an increase of λ. In order to save the computational complexity during encoding process for real-time application, we suggest that a larger value, such as 1/32, of λ should be set. Furthermore, the saving of the computation in software coders corresponds to a reduction of the power consumption in hardware coders, which is also in line with the concept of green computation advocated recently. Moreover, some fast intra [60, 61] or inter [52, 62] techniques can also be combined with this algorithm. Therefore, it gives a possible better selection for real-time applications.

Note that the first algorithm can provide a better rate distortion performance and save more encoding time when compared with other algorithms in the literature. However, the timesaving is fixed. By contrast, the second method is capable of providing a tradeoff between the encoding complexity and the rate distortion performance. Thereby, if we want to save calculation, we will set a large value to λ. Otherwise, if we want to have a better rate distortion performance, a smaller value of λ is preferred.

3.4 Chapter summary

In this chapter, two new algorithms are presented to detect ZQDCT coefficients for inter residual block. In the first algorithm, a new classification of patterns for DCT, Q, IQ and IDCT processes is proposed. Based on the new classification, the quantized DCT coefficients can be coded efficiently. Furthermore, the required thresholds for detecting ZQDCT coefficients are determined by combining the Gaussian distribution with a theoretical analysis of DCT and quantization in H.264/AVC. Experimental results show that the proposed algorithm not only achieves significant time saving for DCT, Q, IQ and IDCT processes, but also shows a better rate-distortion performance than the methods in [6, 17]. It is a possible direction for real-time application.

Moreover, an algorithm based on the Lagrangian FAR-FRR optimization is proposed to detect ZQDCT coefficients for inter MBs. Based on the distribution of residual pixels and the conditional distribution of DCT coefficients, a joint probability model is developed. A Lagrangian FAR-FRR optimization (FFO) method is then proposed to tradeoff between the quality of the reconstructed videos and the encoding complexity. The threshold TN_{uv} for each ZQDCT coefficient is achieved during the process of FFO. Experimental results show that the proposed algorithm can give a better rate distortion performance with a lower encoding complexity compared with the previous methods in the literature.

We would also like to stress that although the two algorithms are proposed based on 4x4 integer DCT, they can also be applied to other forms of DCT, such as floating-point DCT and DCT with a larger size, by a slight modification of the algorithm. Moreover, it is also worth researching to extend this work to other video coding standards based on DCT and quantization technology, such as high efficiency video coding, multi-view video coding or 3D video coding in the future.

Chapter 4 Fast Intra Mode Selection Algorithm in H.264/AVC

4.1 Introduction

In this chapter, we mainly focus on developing a fast algorithm to speed up the intra prediction process in the high profiles of the H.264/AVC standard.

Let us recall that a hierarchical mode decision method for fast intra prediction was found in [25]. In this method, a fixed threshold for the variance of a MB is used to select the block size. This may not be a good arrangement for different quantization step sizes. On the other hand, for the selected block size, a spatial domain filtering [23] (SDF) method is used for mode decision. However, it may not be a good choice since the loss in rate distortion performance is still large based on the results reported in [23] due to using 2x2 pseudo blocks instead of the original MBs/blocks to perform filtering in [23].

In order to further reduce encoding complexity of the intra prediction in H.264/AVC with a less loss in rate distortion performance, we present a new intra mode selection algorithm in this chapter. The proposed algorithm consists of two stages. First, an improved variance based method is used to determine the intra prediction types (I4MB type and I8MB type, or I16MB type and I8MB type). Second, for the selected types, the strength of directional differences (DDS) in the original pixel domain is calculated based on the characteristics of each directional prediction mode, which is defined in the H.264/AVC standard. Better candidate modes are then selected based on the DDS for the rate distortion optimization (RDO) process. Furthermore, an early termination strategy of the RDO process is proposed for the type/mode decision. Experimental results show that the proposed algorithm can achieve more than 80% reduction in encoding time with a negligible loss in rate distortion performance. Details about the proposed algorithm are described as follows.

4.2 Proposed fast intra prediction algorithm for H.264/AVC

In FRExt High profiles of H.264/AVC standard [4], three types, i.e. I4MB type, I8MB type and I16MB type, have to be evaluated for the intra prediction of an MB with the conventional realization. Note that in many cases, it is unnecessary to test all the types. It then becomes a problem of type selection, which is a crucial issue for the success of a fast mode decision.

4.2.1 Selection of candidate intra prediction types

I8MB type is introduced as a compromise between I4MB type and I16MB type in terms of encoding complexity and rate distortion performance in the high profiles. Therefore, the I8MB type is always considered as one candidate type to participate in intra prediction in the proposed algorithm. Generally speaking, I16MB type is suitable for encoding smooth areas of a frame, whereas I4MB type is more efficient to encode regions of a frame, which contain significant details [3]. Therefore, the smoothness of an MB is an extremely important factor for the elimination of one prediction type before RDO calculation. Note that the variance-based method [25] has been proposed to measure the texture complexity of a block. In this work, we extent it to determine the prediction types of an MB. The variance for an MB can be represented as follows.

$$Var = \sum_{i=0}^{15}\sum_{j=0}^{15} f^2(i,j) - \frac{1}{256}\left[\sum_{i=0}^{15}\sum_{j=0}^{15} f(i,j)\right]^2, \tag{4.1}$$

where f(i,j) denotes the luminance value of the pixel at (i,j) in the MB. The criterion proposed in [25] to measure the texture complexity of a block is

$$complexity = \begin{cases} low, & if\ R_B < T \\ high, & otherwise \end{cases},$$

where R_B=ln(Var)/ln(Var_{max}), and Var_{max} is the possible maximum variance of a block. In [25], the threshold T is set to 0.70 according to empirical evaluation for texture classification. As we have mentioned that the intra prediction type has a close relationship with the texture complexity of a block, the variance can also be improved to determine the prediction types of an MB in the proposed algorithm. Thus, the candidate prediction types for the MB can be determined as follows.

$$candidate\ \ types = \begin{cases} I16MB, I8MB, & if\ Var < Th \\ I4MB, I8MB, & otherwise \end{cases},$$

where Th is a threshold used to determine the prediction type of an MB. In [25], the threshold Th is fixed at 92735 for type decision. Generally speaking, a constant threshold is a reasonable choice for texture classification, while it is usually not correct for the determination of an MB encoding type. The reason is that the variance only represents the texture complexity of a block. However, the final encoding type of the MB can be influenced by not only the texture complexity but also the QP value. Figure 4.1 shows the variance distribution of the MBs encoded by different intra prediction types in the first frame of "Salesman" sequence with different QP values. From the results, we can see that the same MB may be encoded using I4MB type, I8MB type, or even I16MB type at different QP values. Thus, Th should not be a constant but be adaptive to QP for the prediction type decision of an MB. In Figure 4.1, the red line indicates the threshold 92735 used in [25] for type decision of an

MB. It is clear that this fixed threshold (for all QP values) is not a good choice since almost all MBs are encoded by I4MB type for a lower QP, while I8MB type and I16MB type are dominantly used to encode MBs with large values of QP.

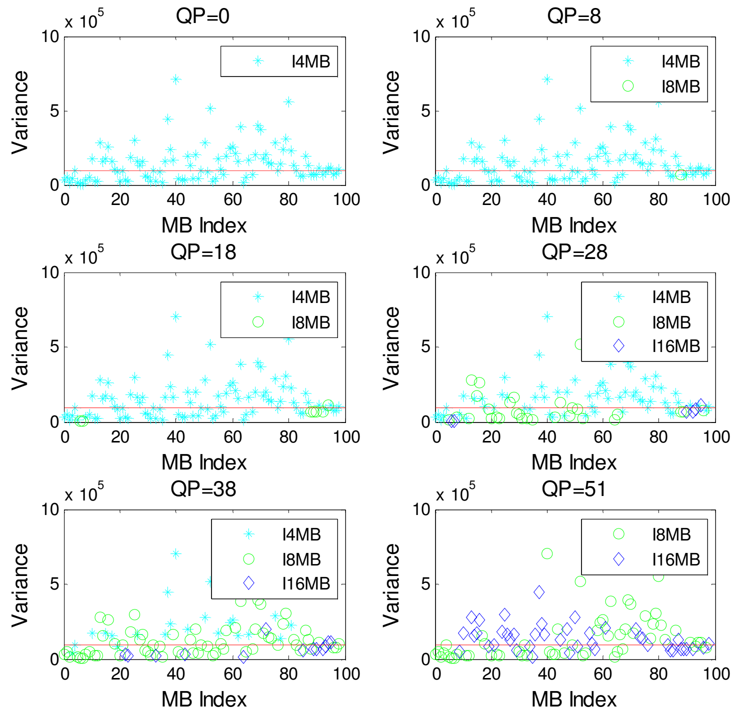


Figure 4.1. Variance distribution of MBs in one QCIF frame

With the observation of the above problem, we propose a variable threshold for different QP values. From the definition of R_B, the threshold, Th, for Var can be found as shown in (4.2). Note that T is no longer a constant but it changes with QP for the determination of intra prediction type in the proposed algorithm. At present, a linear model for T is suggested for simplifying the calculation.

$$Th = e^{T \cdot \ln(Var_{\max})} = e^{(\alpha \cdot QP + \beta) \cdot \ln(Var_{\max})} \tag{4.2}$$

By using (4.1), Var_{max} can be computed as (4.3) for a Luma MB.

$$Var_{\max} = \left[\left(f_{\max}^2 + f_{\min}^2 \right) - \frac{1}{2} \left(f_{\max} + f_{\min} \right)^2 \right] \times \frac{16 \times 16}{2} \tag{4.3}$$

Hence, our key issue is to find the values of α and β for the improved variance based measure (IVM). Observing from extensive encoding results, the following two boundary conditions usually hold:

(i) The encoding type of an MB is not I16MB when the QP has the minimal value: zero.

(ii) The encoding type of an MB is not I4MB when the QP is 51 (which is the maximum value of QP in H.264/AVC standard).

From condition (i), we have

$$\beta \approx 0 \tag{4.4}$$

From condition (ii), we have

$$51\alpha + \beta \geq 1 \tag{4.5}$$

In order to have a better rate distortion performance, we choose the value of α which satisfies the equality part of (4.5). Finally, threshold Th in the proposed algorithm can be derived as

$$Th = e^{T \cdot \ln(Var_{\max})} = e^{\frac{QP}{51} \cdot \ln(Var_{\max})} \tag{4.6}$$

4.2.2 Selection of candidate prediction modes for different types

To further reduce the encoding time when dealing with different prediction types, we propose a directional difference measure (DDM) method based on the intrinsic characteristics of each intra prediction mode defined in H.264/AVC standard. Details are presented as follows.

4.2.2.1 Deciding candidate prediction modes for I4MB/I8MB type

For each 4x4 block, there exist nine prediction modes of I4MB type. According to each intra prediction mode defined in the H.264/AVC standard [1-4], we indicate the locations of pixels, whose predicted values, interestingly, are equal. Figure 4.2

shows these locations where pixels with equal prediction value are labeled using the same color (gray, red, yellow, green, blue and light-blue, but not white).

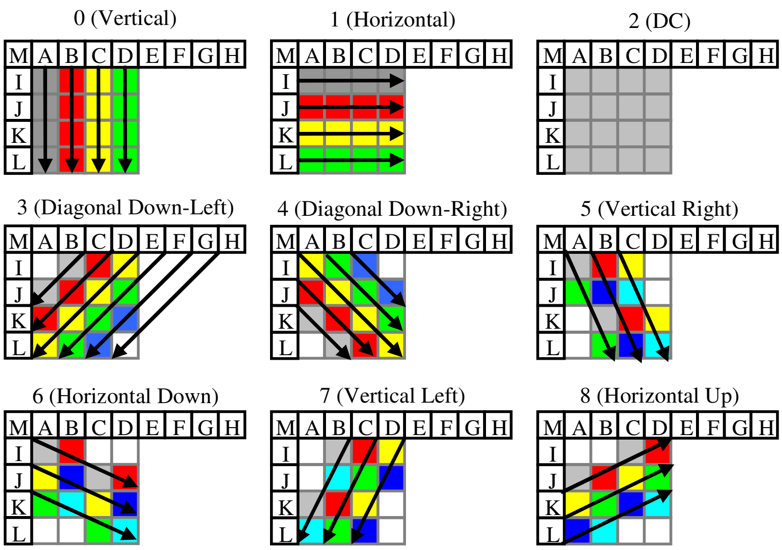


Figure 4.2. Distribution of pixels with equal predicted value in 4x4 Luma intra prediction modes

Intuitively, the nine predicted blocks produced by the nine prediction modes represent nine possible cases of one original block. Therefore, if a block is very similar to one of the nine cases, the residual block between the original block and the predicted block can be very small. As a result, less number of bits is required to encode this residual block. Based on the above observations, a method based on directional difference measure for fast mode decision is presented. Among the nine prediction modes of I4MB type, eight of them are directional modes. For each directional mode, the mean of the absolute differences between two neighboring pixels with the same color in Figure 4.2 is evaluated. The mean value is called the strength of directional differences (DDS) for the corresponding mode in the proposed method. The details are shown as follows.

The DDS for the first four directional modes can be calculated as (4.7)-(4.10) by referring to Figure 4.2.

$$DiffV_{4\times4} = \frac{1}{12}\sum_{x=0}^{3}\sum_{y=0}^{2}\left|f(x, y+1) - f(x, y)\right| \tag{4.7}$$

$$DiffH_{4\times4} = \frac{1}{12}\sum_{y=0}^{3}\sum_{x=0}^{2}\left|f(x+1, y) - f(x, y)\right| \tag{4.8}$$

$$DiffDDL_{4\times4} = \frac{1}{9}\sum_{x=0}^{2}\sum_{y=1}^{3}\left|f(x+1, y-1) - f(x, y)\right| \tag{4.9}$$

$$DiffDDR_{4\times4} = \frac{1}{9}\sum_{y=0}^{2}\sum_{x=0}^{2}\left|f(x+1, y+1) - f(x, y)\right|, \tag{4.10}$$

where (x,y) is the position of the pixel f(x,y) in the current 4x4 block. In fact, the computation method of the first four DDSs also has been proposed in [24]. For the other four directional modes, a linear interpolation method is used to compute their DDS in [24]. For example, the DDS of mode 7 is computed as follows

$$D^7 = \left(D^0 + D^3\right)/2 \tag{4.11}$$

Equation (4.11) indicates that the DDS value D7 of mode 7 must be always in between the DDS value D^0 of mode 0 and the DDS value D3 of mode 3. This results in that the probability, that mode 7 is chosen as the candidate mode, is in between that of mode 0 and mode 3. It is in conflict with the statistical result in H.264/AVC standard. In the H.264/AVC standard, the intra mode with a small index number has a higher probability to be chosen as the best coding mode after RDO calculation, which means that the chosen probability of mode 7 should be smaller than that of mode 0 and mode 3. As a reasonable computation method of DDS, the method itself should not specify the order of the mode. Different from the interpolation method in [24], the DDS for mode 5 to mode 8 can be calculated as (4.12)-(4.15) by referring to Figure 4.2 in our proposed algorithm.

$$DiffVR_{4\times4} = \frac{1}{6}\sum_{y=0}^{1}\sum_{x=0}^{2}\left|f(x+1,\, y+2) - f(x,\, y)\right| \tag{4.12}$$

$$DiffHD_{4\times4} = \frac{1}{6}\sum_{y=0}^{2}\sum_{x=0}^{1}\left|f(x+2,\, y+1) - f(x,\, y)\right| \tag{4.13}$$

$$DiffVL_{4\times4} = \frac{1}{6}\sum_{y=0}^{1}\sum_{x=1}^{3}\left|f(x-1,\, y+2) - f(x,\, y)\right| \tag{4.14}$$

$$DiffHU_{4\times4} = \frac{1}{6}\sum_{y=0}^{2}\sum_{x=2}^{3}\left|f(x-2,\, y+1) - f(x,\, y)\right| \tag{4.15}$$

Table 4.1. Selection Probabilities of some modes

Seq.	QP	Probability of each mode											
		M(0)	M(1)	M(2)	M(3)	M(4)	M(5)	M(6)	M(7)	DC	MPM	LNM	RNM
Coastguard (QCIF)	22	54.230	12.003	6.282	4.609	4.675	4.647	4.254	4.035	5.265	3.456	3.563	6.518
	27	55.421	12.145	6.130	4.443	4.438	4.233	3.977	3.725	5.487	3.961	3.423	6.430
	32	56.606	12.144	6.029	4.567	3.976	3.799	3.431	3.092	6.355	4.981	3.061	5.818
	37	57.639	11.583	5.722	4.619	3.518	3.139	2.591	2.244	8.945	6.270	2.346	4.466
Salesman (QCIF)	22	41.638	14.194	8.997	7.001	5.891	4.719	4.579	3.898	9.083	5.219	4.178	5.063
	27	40.656	14.254	8.536	7.378	5.683	4.558	4.158	3.267	11.510	6.019	4.059	4.559
	32	37.452	14.166	8.672	8.068	5.508	3.938	3.606	2.903	15.688	8.191	3.924	4.075
	37	34.667	13.643	8.808	8.936	5.504	3.738	2.688	1.964	20.053	10.733	3.179	3.167
Football (CIF)	22	44.256	15.401	8.409	5.916	5.945	5.017	4.323	3.569	7.162	5.010	4.554	5.562
	27	42.145	15.892	8.693	6.429	5.875	4.851	4.061	3.275	8.779	6.470	4.482	5.641
	32	37.860	15.040	8.710	7.266	5.398	4.364	3.458	2.712	15.192	8.452	3.721	5.070
	37	28.599	11.796	7.441	7.193	4.025	3.019	2.349	1.832	33.745	9.052	2.254	3.503
Bus (CIF)	22	44.675	11.678	8.228	6.885	5.974	5.213	4.594	3.956	8.798	5.422	3.852	4.416
	27	44.771	11.931	8.451	7.313	5.679	4.668	3.985	3.388	9.813	6.776	3.521	4.148
	32	43.975	12.137	8.632	7.886	5.223	4.010	3.273	2.657	12.207	8.316	3.015	3.612
	37	42.123	11.544	8.327	8.369	4.175	2.770	2.076	1.544	19.173	9.714	2.034	2.477
OldTown Cross (720p)	22	28.341	12.247	8.977	7.616	7.476	6.942	6.513	5.923	15.964	7.976	2.685	3.760
	27	24.661	8.813	5.731	4.719	4.265	3.623	3.168	2.655	42.366	5.275	1.805	2.929
	32	23.929	8.883	5.829	5.120	3.998	3.162	2.604	2.073	44.402	6.506	0.874	1.397
	37	22.952	8.847	6.245	6.181	3.981	2.879	2.185	1.625	45.105	8.861	0.463	0.599
Sunflower (720p)	22	37.105	21.735	11.133	6.368	6.684	4.573	3.630	2.309	6.463	7.868	4.665	4.822
	27	32.906	21.080	11.174	6.571	7.141	5.031	3.990	2.544	9.563	8.930	4.518	5.197
	32	26.281	19.103	10.807	7.224	7.953	5.970	5.362	3.660	13.642	10.954	4.386	5.957
	37	17.911	14.754	8.619	6.998	7.680	6.132	6.742	5.151	26.013	11.814	3.785	6.146
Average		**38.367**	**13.542**	**8.108**	6.570	5.444	4.375	3.817	3.083	**16.699**	**7.343**	3.264	4.389

After the eight DDS values have been computed, the selection of the candidate modes becomes more important. In the proposed algorithm, five candidate modes are suggested. Table 4.1 presents the selection probabilities of some modes based on JM12.2 reference software of the H.264/AVC standard. In the Table 4.1, M(0) to M(7) represent the mode with the minimal DDS to the maximal DDS, respectively. Note that the probability decreases with an increase of the DDS value. According to the statistical results, modes M(0), M(1) and M(2) are firstly selected as three candidate modes. Since DC mode has no direction, it usually represents smooth areas compared with the eight directional modes. The statistical results of DC mode in Table 4.1 also show that it has a very high probability of being used. Hence it is also included in the proposed algorithm as the fourth candidate mode to be tested. In addition, the selection probabilities of the most probable mode (MPM) are investigated. As we all know that, if the prediction mode of the current block is the same as the most probable mode, which is defined as the prediction mode of the upper block or the left block as long as it has a smaller mode number, only one bit is required to encode the prediction mode for the current block making use of the CAVLC entropy coding method of the JM reference software in H.264/AVC. Since the neighboring modes in direction are often considered as the candidate modes in many fast mode decision algorithms, such as [22, 23, 25, 61], the chosen probabilities of two directional-neighboring modes (LNM and RNM, which are the modes of the left neighbor and the right neighbor of the prediction direction in the H.264/AVC, respectively) of the mode with the minimal DDS are also computed. Note also that the MPM may be among the first four candidate modes; and the LNM or RNM may also be among M(0), M(1) and M(2). Therefore, the selection probabilities of MPM, LNM and RNM excluding their overlapped modes in the first four candidate modes are also listed in Table 4.1. Based on the results, the MPM is selected as the fifth candidate mode. However, if the MPM belongs to one of the other four candidate modes, the fifth candidate mode is the mode with the fourth minimal DDS.

For I8MB type, the selection method of the candidate modes is the same as that for I4MB type.

4.2.2.2 Deciding candidate prediction modes for I16MB type and two Chroma blocks

For I16MB type, there are only two mathematically directional prediction modes of the I16MB type: vertical mode and horizontal mode. Similar to $DiffV_{4x4}$ and $DiffH_{4x4}$ of the I4MB type, the DDS of the two prediction modes: $DiffV_{16x16}$ and $DiffH_{16x16}$, can be derived with reference to (4.7) and (4.8).

$$DiffV_{16\times16} = \frac{1}{210}\sum_{x=0}^{15}\sum_{y=0}^{14}\left|f(x, y+1) - f(x, y)\right| \tag{4.16}$$

$$DiffH_{16\times16} = \frac{1}{210}\sum_{y=0}^{15}\sum_{x=0}^{14}\left|f(x+1, y) - f(x, y)\right| \tag{4.17}$$

For the plane prediction mode in I16MB type, a linear ‘plane’ function, which is defined in (4.18), is used to compute the predicted values for the samples in an MB. This prediction works well in areas of smoothly-varying luminance [3].

$$p(x, y) = Clip((a + b\times(x-7) + c\times(y-7) + 16) >> 5), \tag{4.18}$$

where p(x,y) denotes the predicted value of pixel f(x,y). Clip(•) is the operator to make the prediction pixel value be limited between 0 and 255; a, b and c are constants depending on its above and left reconstructed pixel values of the current MB [3].

In order to calculate the DDS of this prediction mode, we define two pseudo blocks, PseudoBlockV and PseudoBlockH, as follows.

$$PseudoBlock_V(x, y) = \left|p(x, y+1) - p(x, y)\right| \tag{4.19}$$

$$PseudoBlock_H(x, y) = |p(x+1, y) - p(x, y)| \tag{4.20}$$

Combined with (4.18), we can see that PseudoBlockH and PseudoBlockV have smooth rows and smooth columns, respectively. Hence the DDS for the two pseudo blocks can be given by (4.21) and (4.22), respectively.

$$DiffPseB_V = \frac{1}{224} \times \sum_{x=0}^{15} \sum_{y=0}^{13} |PseudoBlock_V(x, y+1) - PseudoBlock_V(x, y)| \tag{4.21}$$

$$DiffPseB_H = \frac{1}{224} \times \sum_{x=0}^{13} \sum_{y=0}^{15} |PseudoBlock_H(x+1, y) - PseudoBlock_H(x, y)| \tag{4.22}$$

The DDS of the plane prediction mode can also be calculated by (4.23).

$$DiffPlane_{16x16} = \frac{1}{2} \times (DiffPseB_V + DiffPseB_H) \tag{4.23}$$

Finally, the mode with the minimal DDS is chosen as one of two candidates. To achieve a good prediction performance in a smooth block or a boundary MB, the DC mode is also chosen as the other candidate mode.

For the two 8x8 Chroma blocks (U and V) corresponding to each Luma MB, the prediction modes are almost the same as those of I16MB type except for the differences in block size and mode numbers [3]. This makes the measure method of DDS used in I16MB type be applicable here, and the subsequent procedures to predict two candidate modes are also the same as those in I16MB type. After two candidate modes have been determined, the Hadamard SAD algorithm [5] is used to select a better one from the two candidate modes for two Chroma components.

4.2.3 Early termination strategy for RDO calculation

It is well known that RDO technique plays an important part on improving the rate distortion performance of video encoders. However, it also contributes most of the complexity of an encoder based on RDO technique. In the past few years, many fast algorithms have been proposed to speed up the RDO process of mode decision, such as [22-25, 61, 63]. Note that all these algorithms can reduce the complexity at the cost of degradation in rate distortion performance. In this chapter, we develop a new method based on an early termination strategy of RDO process for fast mode decision. This method does not cause any loss of rate distortion performance compared with the original method used in H.264/AVC standard.

In the JM reference software [5] of H.264/AVC, (4.24) is used as the rate-distortion cost function to determine the best one from all the candidate modes.

$$J(MODE) = SSD(s, c, mode \mid QP) + \lambda \cdot R(s, c, mode \mid QP), \tag{4.24}$$

where SSD(s,c,MODE|QP) represents the sum of the squared difference between the original block, s, and its reconstructed block, c. λ is the Lagrange multiplier for mode decision. For I-slice, MODE represents one of the nine intra prediction modes for I4MB/I8MB. R is the number of bits for coding mode index and the quantized DCT (QDCT) coefficients of each block by using the entropy coding method. J denotes the cost that corresponds to one prediction mode for the current block.

Obviously, for each 4x4/8x8 intra coded block, there always exists one mode that can give the best prediction such that the resultant cost J is the smallest. In contrast, the other modes could result in either large D values or larger R values due to the poor prediction. When D or R is larger than the minimal J of the best mode among the modes which have been checked, the RDO process can be stopped, and thereby the corresponding coding process R or D can be skipped. In order to determine which term between D and R should be firstly computed, let us examine term D and term R in (4.24) carefully. For each 4x4 block, we can have nine D values and nine R values due to nine prediction modes in I4MB type. Note that the variance can be used as a measure of how far a set of numbers are spread out from each other. Therefore, we computed the variance of nine D values and the variance of nine R values for each 4x4 intra coded block. Fig. 4 shows the distribution of the variances for D and R in one frame. From this figure, we can see that the variance of D is usually larger than the variance of R. Note also that the variance of term D is larger with an increase of QP as shown in Figure 4.3. This means that the nine D values usually fluctuate more widely than the nine R values for each 4x4 block, and the fluctuation of term D is more widely with an increase of QP. Different from the strategy in [64], where R is computed firstly, we suggest that the distortion term in (4.24) should be calculated firstly based on our investigation. It is then compared with the current minimal rate distortion cost J. If it is larger than the current minimal rate distortion cost, the entropy coding process used to compute the rate term in (4.24) can be skipped.

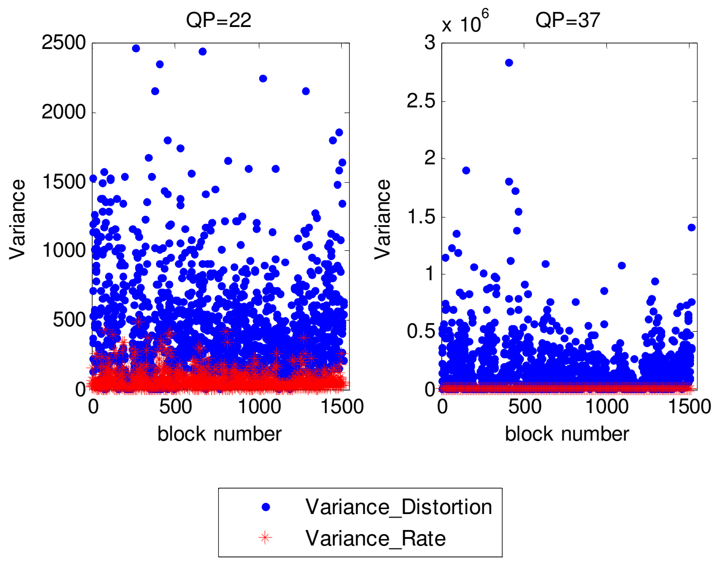


Figure 4.3. Distribution of the variances of D/R for all 4x4 blocks in one intra coded frame

Specifically, the entire early termination procedure is outlined as follows.

Block Level: (MODE: Vertical, Horizontal,...Horizontal Up)

For each 4x4/8x8 block:

Initialize min_rdcost

Compute J(mode 0) by summing SSD(mode 0) and λR(mode 0);

Set min_rdcost = J(mode 0);

For each mode i (i=1,2,...8)

Compute SSD(mode i) by performing DCT, Q, IQ, IDCT of the residual block;

If SSD(mode i) < min_rdcost then

Perform entropy coding of some information associated with mode i to compute $R(mode_i)$;

Compute J (mode i) by summing SSD(mode i) and λR(mode i);

If J (mode i)< min_rdcost **then**

min_rdcost = J (mode i);

End if

else

Skip entropy coding process, and go to the next mode until all the prediction modes have been checked;

End If

End for

Note that the above table describes the early termination of RDO process on a block level basis. Similarly, the early termination method can also be applied to the mode decision on an MB level basis. For MB level, the possible modes include: (a) I4MB, I8MB, I16MB for I-slices, (b) inter modes (SKIP, 16x16, 16x8, 8x16, Tree8x8) and intra modes (I4MB, I8MB, I16MB) for P-slices, and (c) more modes for B-slices [1]. In the original RDO calculation, entropy coding is performed once for every mode. However, with the termination strategy, only part of the modes are necessary to go through the entropy coding process. This early termination strategy is especially efficient for an inter frame coding. In an inter frame, the chance of selecting the intra mode for an MB is only 2.94% on average, but there could be about five times more RDO calculations than that for inter modes [51]. This results show that intra modes usually result in poor prediction compared with motion estimation for the current MB/block. As a result, the distortion term SSD corresponding to an intra mode is usually larger than the current minimal rate distortion cost. Thus the entropy coding process can be terminated. Note that the term R in (4.24) for the MB level includes not only the number of bits used to encode the QDCT coefficients, but also includes the number of bits used for encoding the MB prediction type, the transform size, the entropy coding method and so on. As a result, with the early termination method, the encoding time can be reduced significantly without leading to the degradation of video quality compared with the JM reference software of H.264/AVC standard.

Table 4.2. Statistical results of skip ratio in P frames

Seq.	QP	**Skip Percentage in P frames (%)**		
		I4MB	I8MB	I16MB
Monitor (QCIF)	23	45.533	53.318	56.391
	28	68.106	69.207	68.570
	33	74.942	73.005	69.907
	38	74.234	74.009	68.855
Children (CIF)	23	51.006	60.728	53.527
	28	55.535	63.416	58.783
	33	57.721	63.142	61.917
	38	58.837	61.895	64.504
ShuttleStart (720p)	23	30.372	16.059	27.334
	28	36.518	31.402	26.445
	33	38.279	29.005	21.205
	38	39.285	31.174	19.317

Note also that the more the number of the potential modes (which require RDO calculation) is, the more efficient the early termination method is. Table 4.2 lists the statistical results of the three intra types in which the entropy coding process is skipped in 149 P frames of each video sequence. Another point that we have to stress is that this strategy can work with any fast mode decision algorithms based on RDO process and no additional computation is required. Moreover, no loss in terms of RD performance can be resulted in compared with conventional algorithms.

4.3 Experimental Results

The proposed algorithm has been realized using the H.264/AVC reference software JM 12.2 [5] encoder provided by JVT for the performance evaluation. The test was based on the high profiles with specifications provided in [59]. The GOP structures used in the experiments are III and IPPP. The video sequences with different sizes were used in the test. A total of 150 frames were used for each sequence. The average value of PSNR($\overline{PSNR}$) for Luma and Chroma components was computed according to [22]. The average changes of $\overline{PSNR}$ (ΔPSNR) and bitrate (ΔBR) were computed using the method in [50]. The reduction of encoding time (ΔTIME) with respect to the full search algorithm in [5] was also measured. Four QP values: 22, 27, 32 and 37, were used for I frame. For inter frame, 23, 28, 33 and 38 were used. In the following tables, positive values mean increase, and negative values mean decrease.

4.3.1 Experimental results on all intra coded frames

4.3.1.1 Experimental results of the fast intra type selection algorithms

Table 4.3 lists the evaluation results of the method in [25] and the IVM method for the decision of candidate types in terms of ΔPSNR, ΔBR and ΔTIME. From the results, we can see that the proposed algorithm using a variable threshold to determine the prediction type for an MB can achieve a better rate distortion performance compared with the fixed threshold used in [25] for all tested video sequences. On average, 0.018dB loss in PSNR and 0.295% bitrate increase are found by IVM strategy, while 0.033dB loss and 0.560% increase are resulted in by the strategy [25] in terms of PSNR and bitrate, respectively. Therefore, we can see that the proposed strategy using a variable threshold to determine the prediction type for a MB can achieve a better rate distortion performance compared with the fixed threshold used in [25]. This well illustrates our analysis on Figure 4.1 in Section 4.2.1.

Another observation is that the timesaving of the method [25] is almost the same for the four QP values of each video sequence as shown in Table 4.3. For example, the timesaving is about 20% for the four observed QP values of the "Monitor" sequence. The reason can be attributed to only one threshold used in [25] to decide the candidate types for all QP values. As a result, the candidate types are the same for all cases no matter what QP is. This strategy can result in two problems. One is that in case of small QP values, the fixed threshold may result in poor rate distortion performance. On the contrary, the method may require much time using the same threshold when the QP value is larger. As for our strategy using an adaptive threshold, it can save more encoding time as shown in Table 4.3 with an increase of QP since a larger threshold is used for a larger QP based on (4.6). For example, the timesaving increases from 7.961% to 21.830% when QP is from 22 to 37 for the "Monitor" sequence. When the QP value is equal to 37, the timesaving obtained by the two methods is almost the same as that in [25]. Thus we can arrive at a conclusion that the IVM method should be more efficient than that in [25] in terms of timesaving when the QP is larger than 37 since a lager threshold is used.

Table 4.3. Evaluation results of the proposed IVM algorithm

Seq.	QP	Previous algorithm [25]			IVM algorithm		
		Δ TIME [%]	Δ PSNR [dB]	Δ BR [%]	Δ TIME [%]	Δ PSNR [dB]	Δ BR [%]
Monitor (QCIF)	22	-20.199	-0.075	1.047	-7.961	-0.047	0.658
	27	-19.678			-13.128		
	32	-22.299			-16.771		
	37	-22.739			-21.830		
Mobile (QCIF)	22	-5.061	-0.020	0.185	-6.791	-0.003	0.045
	27	-6.809			-7.545		
	32	-7.740			-7.608		
	37	-7.445			-7.217		
Container (CIF)	22	-23.496	-0.051	0.766	-7.159	-0.042	0.637
	27	-22.991			-11.469		
	32	-23.651			-21.419		
	37	-23.577			-24.182		
Bus (CIF)	22	-19.082	-0.014	0.184	-6.918	-0.004	0.051
	27	-18.869			-8.115		
	32	-18.709			-10.616		
	37	-18.859			-18.032		
Intotree (720p)	22	-39.224	-0.023	0.672	-9.472	-0.012	0.366
	27	-30.294			-17.478		
	32	-39.651			-24.005		
	37	-47.196			-46.567		
Shuttlestart (720p)	22	-46.628	-0.017	0.507	-20.515	-0.000	0.010
	27	-46.732			-25.214		
	32	-57.321			-50.954		
	37	-59.014			-55.019		
Average	\	\	-0.033	0.560	\	-0.018	0.295

Although the timesaving achieved by IVM (the first step of our algorithm) is less than that in [25] for the four QP values, it does not give influence to the success of the whole algorithm (IVM + DDS + Early termination of RDO calculation) proposed by us. This can be illustrated by Tables 4.4-4.6. Furthermore, the proposed algorithm gives a better rate distortion performance compared with other algorithms.

4.3.1.2 Experimental results of the fast intra type and mode decision algorithms

Tables 4.4-4.6 summarize the simulation results achieved by the proposed and the previous fast algorithms [23, 25] in terms of three measures: ΔPSNR, ΔBR and ΔTIME. It can be observed from these three tables that, by combining IVM, DDM and early termination of RDO calculation, the proposed algorithm can save more than 80% encoding time compared with JM12.2 for all test video sequences, with a slight decrease in RDO performance. When comparing the proposed algorithm with the previous algorithms [23, 25], we find that the proposed algorithm obviously outperforms the previous algorithms among all the three measures for all the intra coded sequences. On average, less than 0.2dB loss in PSNR or 2.5% bitrate increase is resulted from the proposed algorithm. By contrast, the average loss of PSNR in [23] or [25] is more than 0.2dB, and a larger increase in bitrate (more than 3%) can also be observed in both [23] and [25].

Table 4.4. Evaluation results of the proposed algorithm for QCIF sequences

Seq.	**Previous Algorithm [23]**			**Previous Algorithm [25]**			**The proposed algorithm**		
	Δ PSNR [dB]	Δ BR [%]	Δ TIME [%]	Δ PSNR [dB]	Δ BR [%]	Δ TIME [%]	Δ PSNR [dB]	Δ BR [%]	Δ TIME [%]
Coastguard	-0.225	3.411	-61.514	-0.231	3.464	-70.955	-0.115	1.724	-82.794
Monitor	-0.302	4.224	-63.336	-0.373	5.270	-71.051	-0.148	2.072	-82.874
Foreman	-0.149	2.437	-59.361	-0.192	3.158	-64.880	-0.134	2.182	-82.388
Mobile	-0.182	1.737	-59.608	-0.200	1.913	-63.144	-0.181	1.726	-81.787
Salesman	-0.217	3.367	-61.731	-0.271	4.208	-71.181	-0.136	2.174	-82.989
Average	**-0.215**	**3.035**	**-61.110**	**-0.253**	**3.603**	**-68.242**	**-0.143**	**1.976**	**-82.566**

Table 4.5. Evaluation results of the proposed algorithm for CIF sequences

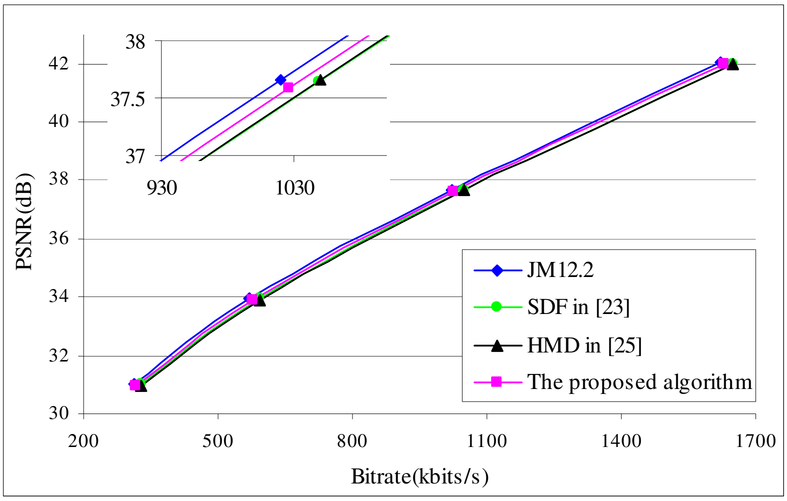
Seq.	**Previous Algorithm [23]**			**Previous Algorithm [25]**			**The proposed algorithm**		
	Δ PSNR [dB]	Δ BR [%]	Δ TIME [%]	Δ PSNR [dB]	Δ BR [%]	Δ TIME [%]	Δ PSNR [dB]	Δ BR [%]	Δ TIME [%]
Children	-0.205	2.533	62.277	-0.244	3.031	-69.290	-0.220	2.717	-83.108
Container	-0.206	3.162	-61.262	-0.278	4.289	-70.593	-0.220	3.41	-83.827
Hall	-0.286	5.228	-62.646	-0.372	6.832	-72.038	-0.178	3.236	-83.924
Mobile	-0.168	1.681	-59.922	-0.186	1.865	-64.318	-0.156	1.555	-82.972
Bus	-0.247	3.226	-62.776	-0.263	3.439	-70.664	-0.124	1.617	-83.472
Average	**-0.222**	**3.166**	**-61.777**	**-0.269**	**3.891**	**-69.381**	**-0.180**	**2.507**	**-83.461**

Table 4.6. Evaluation results of the proposed algorithm for 720p sequences

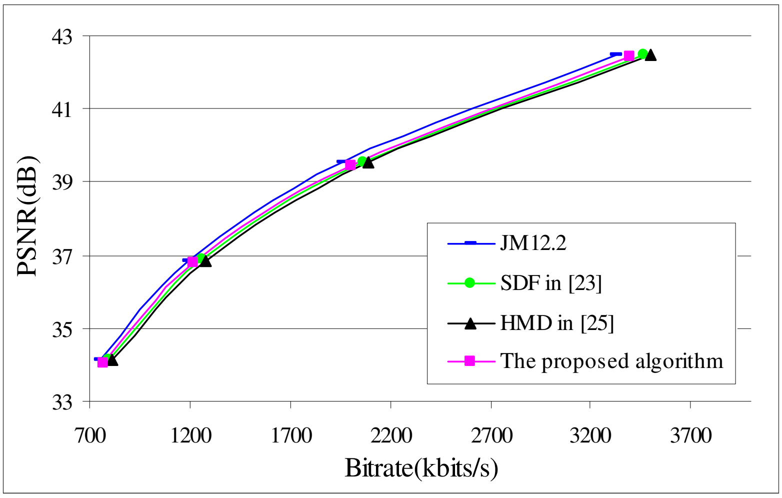
Seq.	Previous Algorithm [23]			Previous Algorithm [25]			The proposed algorithm		
	Δ PSNR [dB]	Δ BR [%]	Δ TIME [%]	Δ PSNR [dB]	Δ BR [%]	Δ TIME [%]	Δ PSNR [dB]	Δ BR [%]	Δ TIME [%]
CrowdRun	-0.136	1.972	-59.420	-0.154	2.242	-65.522	-0.143	2.079	-84.160
OldTown	-0.168	3.865	-59.522	-0.202	4.642	-70.828	-0.115	2.642	-84.199
Shuttlestart	-0.290	8.763	-62.376	-0.301	9.092	-79.214	-0.125	3.742	-87.666
Sunflower	-0.202	3.834	-62.730	-0.209	3.959	-78.458	-0.190	3.604	-84.929
Average	**-0.199**	**4.609**	**-61.012**	**-0.217**	**4.984**	**-73.506**	**-0.143**	**3.017**	**-85.239**

Figure 4.4 shows the rate-distortion curves for three sequences with different sizes. These Figures further illustrate that the proposed algorithm gives the best RD performance compared with other algorithms.

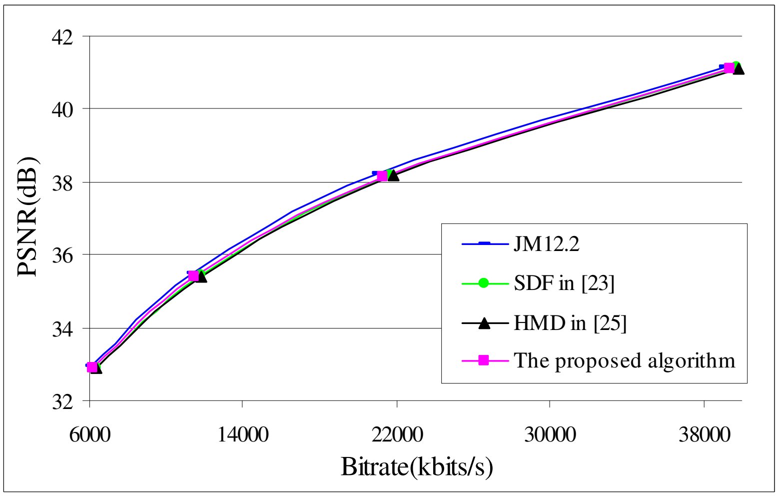
Note further that although one more candidate mode in the proposed algorithm is used compared with most of other fast intra mode decision algorithms [22-25], it does not increase the overall encoding time, since a good termination strategy for the RDO process is adopted. As a result, the RD performance is also high since a larger pool of candidate modes has a larger probability to include the best prediction mode.



(a) Coastguard. QCIF



(b) Hall. CIF



(c) OldTownCross. HDTV (720p)

Figure 4.4. Rate-distortion curves for three sequences with different sizes

4.3.2 Experimental results on IPPP prediction structure

In the H.264/AVC video coding standard, intra coding types and modes are also available in P-frames. Hence, MBs in a P frame also need to perform the intra coding process. Since the effort of computing the rate-distortion costs of the intra mode is about five times higher than that for inter modes [51], a fast and precise intra mode selection algorithm is also desirable for encoding MBs in P frames. Hence, we also applied the proposed algorithm (IVM + DDS + Early termination of RDO calculation) to the IPPP structure in order to evaluate its performance. In our test, one intra frame followed by 149 P frames are set for all video sequences. Tables 4.7-4.9 show the simulation results.

Table 4.7. Evaluation results on IPPP prediction structure for QCIF sequences

Sequence	Previous Algorithm [25]			The proposed algorithm		
	Δ PSNR [dB]	Δ BR [%]	Δ TIME [%]	Δ PSNR [dB]	Δ BR [%]	Δ TIME [%]
Coastguard	-0.034	0.803	-28.060	-0.033	0.776	-33.238
Monitor	-0.078	1.634	-26.816	-0.052	1.180	-31.481
Carphone	-0.008	0.181	-26.554	0.004	-0.089	-30.999
Mobile	-0.018	0.335	-29.019	-0.005	0.079	-37.903
Salesman	-0.081	1.716	-27.859	-0.060	1.229	-33.418
Average	**-0.044**	**0.934**	**-27.662**	**-0.029**	**0.635**	**-33.408**

Table 4.8. Evaluation results on IPPP prediction structure for CIF sequences

Sequence	Previous Algorithm [25]			The proposed algorithm		
	Δ PSNR [dB]	Δ BR [%]	Δ TIME [%]	Δ PSNR [dB]	Δ BR [%]	Δ TIME [%]
Paris	-0.029	0.545	-28.897	-0.028	0.486	-35.033
Bus	-0.021	0.393	-28.990	-0.013	0.231	-34.563
Crew	-0.082	1.994	-27.102	-0.070	1.679	-30.387
Flower	-0.022	0.360	-26.379	-0.017	0.267	-35.103
Silent	-0.053	1.293	-26.840	-0.040	0.985	-31.715
Average	**-0.041**	**0.917**	**-27.642**	**-0.034**	**0.730**	**-33.360**

Table 4.9. Evaluation results on IPPP prediction structure of HDTV sequences

Sequence	Previous algorithm [25]			The proposed algorithm		
	Δ PSNR [dB]	Δ BR [%]	Δ TIME [%]	Δ PSNR [dB]	Δ BR [%]	Δ TIME [%]
Crowdrun	-0.04	0.303	-26.796	-0.014	0.300	-32.243
Oldtown	-0.030	1.278	-25.027	-0.025	1.032	-28.919
Shuttlestart	-0.080	3.098	-22.310	-0.066	2.574	-24.507
Sunflower	-0.055	1.316	-26.255	-0.053	1.262	-27.540
Average	**-0.051**	**1.499**	**-25.097**	**-0.040**	**1.292**	**-28.302**

From these tables, we can see that the proposed algorithm can achieve a better rate distortion performance compared with algorithms like reference [25]. It also can be observed that, the proposed algorithm can achieve consistent timesaving of more than 30% on average, In a practical application, we may also combine the proposed algorithm with any fast inter mode decision algorithms in the literature [51, 52, 62, 65, 66], which can achieve a high-speed and quality video coding with the IPPP structure.

4.4 Chapter summary

In this chapter, we propose some techniques to accelerate the process of intra prediction. First, an improved variance-based method is proposed to determine the intra prediction types for an intra coded MB. Different from simple texture complexity classification, an adaptive threshold is suggested based on the quantization parameter. This makes the threshold more reasonable, and thereby the determination of intra prediction type is more accurate. Another strategy is to make full use of the directional difference measure of each prediction mode such that the better candidate prediction modes can be selected accurately. This strategy critically touches upon the intrinsic property of the way H.264/AVC makes the intra mode prediction. Furthermore, an early termination strategy of RDO process is proposed, which can be applied to any mode decision algorithms based on the RDO technique. Especially for type decision in P frames, most of the unnecessary entropy coding corresponding to intra types can be skipped. All these strategies can efficiently simplify the process of intra prediction mode selection. Experimental results show that the proposed algorithm can achieve more than 80% timesaving used in the JM12.2 reference software at the expenses of a slight decrease in PSNR and a slight increase in bit rate. Compared with other algorithms in the literature, the proposed algorithm obviously shows the best performance among all the three measures.

Chapter 5 Improved Lossless Intra Coding Algorithms for videos in H.264/AVC

5.1 Introduction

Note that the H.264/AVC has a block-based coding structure [1-4], samples away from the references are not predicted well due to poor correlation. The compression efficiency is thus limited, especially for lossless intra coding without transform and quantization operations. In order to improve the lossless intra coding algorithm in H.264/AVC, two algorithms are proposed in this chapter.

Let us recall that the samplewise DPCM instead of using a block-based approach are presented in [26] and [27]. In [26], the samplewise DPCM approach is firstly suggested for lossless intra prediction. For adopting near samples as reference for prediction, the compression efficiency is enhanced. This algorithm was adopted in the H.264/AVC standard. However, only the modes with one sample predictor can be improved by using the samplewise DPCM method in [26]. As a result, the performance is limited. In [27], the authors extended the sample DPCM concept to all intra prediction modes by using linear interpolation. Hence, a higher compression efficiency is obtained.

To further improve the coding efficiency, two efficient lossless intra coding algorithms based on H.264/AVC framework are presented in this chapter. In the first proposed algorithm, two contributions have been made. One is that samples in an MB/block are hierarchically predicted instead of using block-based prediction as a whole. More specifically, four groups are extracted from the samples in an MB/block. Samples in the first group are firstly predicted based on directional intra prediction method, and then the samples in other groups are predicted using the samples in the first group as the references. As a result, the information left in the residual block can be reduced since the samples can be predicted accurately by using nearer references. The other contribution is that two coding modes: one-layer coding (OLC) mode and two-layer coding(TLC) mode are designed to encode efficiently the resulting residual block. A better coding mode can be selected based on the rate optimization. Experimental results show that the proposed algorithm can provide a better compression efficiency than lossless intra coding in JM12.2.

Note that the adaptive interpolation filter achieves a significant contribution in the formulation of the sub-pixel reference frame for motion estimation and compensation due to its adaptation. Inspired by its advantages, we make use of the adaptive interpolation filter to further improve lossless intra coding in H.264/AVC to form our second algorithm. In the second algorithm, a hierarchical intra prediction algorithm combined with an adaptive interpolation filter is proposed to improve the lossless intra prediction. Similar to the first algorithm, four subblocks: G0, G1, G2 and G3 are firstly formed by sampling pixels in an MB/block, and then a hierarchical intra prediction is performed on these subblocks. Samples in G0 are predicted based on the spatial prediction method in H.264/AVC, and subsequently are encoded and reconstructed. Samples in G1, G2 and G3 are then predicted based on an adaptive interpolation filter by using the neighboring reconstructed samples as predictors. The filter coefficients for each subblock are estimated adaptively in a training window (G0 or G1) based on the geometric duality. Note that the adaptive interpolation filter has the ability to adapt locally to varying pixel structures (such as textures or edges).

As a result, a desirable prediction is obtained due to the use of neighboring samples as reference and the adaptation of the interpolation filter. The two coding modes: OLC and TLC are then used to encode the residual blocks with different distributions for efficient entropy coding. Experimental results show that about 13.485%, 11.803% and 13.135% of bitrate saving on average are achieved for the tested QCIF, CIF and HDTV sequences compared with JM18.0, respectively. When compared with other algorithms in the literature, the proposed algorithm also shows the best compression efficiency.

5.2 Lossless intra coding algorithm based on hierarchical intra prediction and residual coding optimization in H.264/AVC

To obtain a better prediction for samples away from their references, a hierarchical intra prediction (HIP) method is presented in this section. The details about the HIP method are given from two aspects: the HIP methods for the I8MB type, and the I16MB type/Chroma intra prediction. Furthermore, a coding scheme selection (CSS) method is proposed to efficiently encode the resulting residual by the HIP method.

5.2.1 HIP method for I8MB type

In FRExt profiles, I8MB type is introduced to improve the encoding efficiency of the intra prediction in H.264/AVC standard. An MB is firstly divided into four 8x8 blocks. For each 8x8 block, nine prediction modes are used to do intra prediction. In order to improve the prediction accuracy for samples away from their references, we suggest that samples in each 8x8 block be predicted hierarchically instead of using the whole-block based method in the original H.264/AVC standard. As a result, four groups from G0 to G3 in each 8x8 block are classified as shown in Figure 5.1.

By sampling with the corresponding group numbers as shown in Figure 5.1 (a), four 4x4 blocks are formed as shown in Figure 5.1 (b) for the residual entropy coding process.

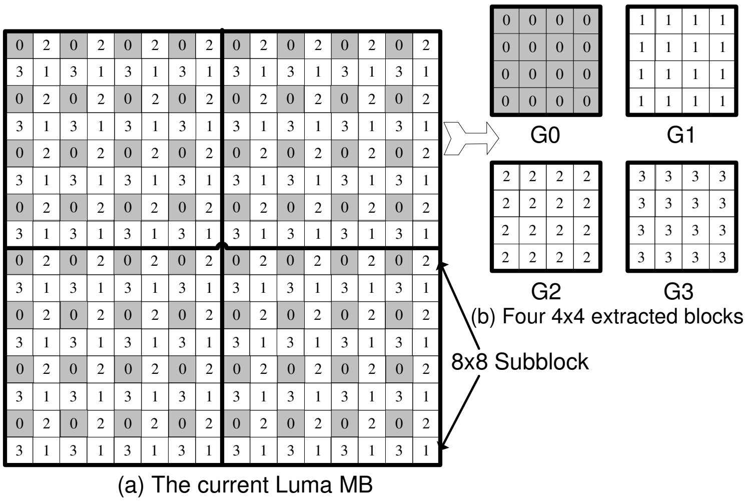


Figure 5.1. Structure of HIP for I8MB type

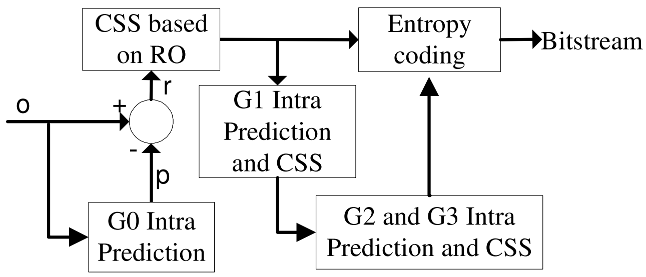


Figure 5.2. Block diagram of the proposed HIP

Figure 5.2 shows a block diagram of the proposed HIP for an 8x8 block, where *o* denotes an original block G0, *p* denotes the predicted block of G0, and *r* is the difference between *o* and *p*. The detailed prediction strategies of samples in the four groups are presented as follows:

First, select the best prediction mode for samples in G0 from nine prediction modes of the I8MB type. Each 8x8 block in an MB is predicted as a whole according to the nine prediction modes of the I8MB type in H.264/AVC standard. However, only the predicted values of the 16 samples labeled as 0 are computed based on a samplewise DPCM method instead of using all 64 samples. After the samples in G0 are predicted from their reconstructed neighbors using one prediction mode, and the residual block r can then be obtained by subtracting the predicted block p from the original block o. For the residual block r, two coding schemes: one-layer coding scheme and two-layer coding scheme, are designed to encode it. The rate costs (J_{OLC} and J_{TLC}) corresponding to the two coding schemes can be given by (5.7) and (5.8), respectively. The coding scheme with the smaller rate cost is selected as the final coding scheme for the residual block r, and the corresponding rate cost is chosen as the rate of this prediction mode as shown in (5.1). The details about the two coding schemes will be illustrated in part D of this section.

$$J(MODE) = \begin{cases} J_{OLC}, & \text{if } J_{OLC} < J_{TLC} \\ J_{TLC}, & \text{otherwise} \end{cases}, \tag{5.1}$$

where MODE represents one of the nine intra prediction modes of I8MB type.

By using (5.1), each of the nine residual blocks produced by the nine prediction modes for G0 has a smaller coding cost. Finally, the prediction mode with the minimum coding cost is selected as the prediction mode of G0. Since it is a lossless coding process, there is no degradation in PSNR. The samples can be reconstructed without any loss of fidelity. As a result, the mode with the minimum rate can make the best compression efficiency for an encoder based on the H.264/AVC framework.

Second, make intra prediction for the samples in G1,G2 and G3 based on a measure of the intensity gradient. The remaining 48 pixels in the 8x8 block are classified into three groups (G1, G2 and G3), which correspond to three 4x4 blocks as shown in Figure 5.1 (b). Samples in G1 are firstly predicted using samples in G0 as references. The predicted values of samples can be obtained adaptively by (5.2) based on a measure of the gradients along different directions (see Figure 5.3), and then the residual block can be obtained by subtracting the predicted G1 from the original G1. Similar to the coding method of the residual blocks for G0, two coding schemes: one-layer coding scheme and two-layer coding scheme, are adopted to encode the residual block of G1. The only difference of the residual coding schemes between G0 and G1 is that there is only one residual block for G1 since the predicted values of samples in G1 are obtained adaptively based on a measure of the gradients not based on nine prediction modes. A better coding scheme for the residual block of G1 is thus selected by comparing J_{OLC} with J_{TLC}. Since it is a lossless coding process, G1 can be reconstructed perfectly. Subsequently, samples in G2 and G3 can be predicted using samples in G0 and G1. The predicted values of samples in the G2 and G3 are also obtained adaptively based on the measure of the gradients along different directions. Finally, the coding scheme for the residual block of G2 or G3 can be determined using the method similar to G1. Figure 5.3 shows the predicted structures of samples in G1, G2 and G3. Referring to this figure, the predicted values of samples in G1, G2 and G3 are calculated by (5.2)-(5.4), respectively.

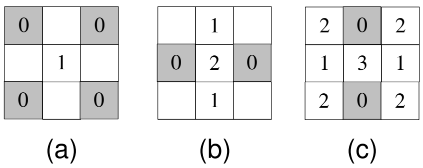


Figure 5.3. Predicted structures of samples in G1, G2, and G3

$$p_{i,j}^{1}=\begin{cases}\left(f_{i-1,j-1}^{0}+f_{i+1,j+1}^{0}+1\right)>>1, & \left|f_{i-1,j-1}^{0}-f_{i+1,j+1}^{0}\right|<\left|f_{i+1,j-1}^{0}-f_{i-1,j+1}^{0}\right| \\ \left(f_{i+1,j-1}^{0}+f_{i-1,j+1}^{0}+1\right)>>1, & otherwise\end{cases} \tag{5.2}$$

$$p_{i,j}^{2}=\begin{cases}\left(f_{i-1,j}^{0}+f_{i+1,j}^{0}+1\right)>>1, & \left|f_{i-1,j}^{0}-f_{i+1,j}^{0}\right|<\left|f_{i,j-1}^{1}-f_{i,j+1}^{1}\right| \\ \left(f_{i,j-1}^{1}+f_{i,j+1}^{1}+1\right)>>1, & otherwise\end{cases} \tag{5.3}$$

$$p_{i,j}^{3}=\begin{cases}\left(f_{i-1,j}^{1}+f_{i+1,j}^{1}+1\right)>>1, & \left|f_{i-1,j}^{1}-f_{i+1,j}^{1}\right|<\left|f_{i,j-1}^{0}-f_{i,j+1}^{0}\right| \\ \left(f_{i,j-1}^{0}+f_{i,j+1}^{0}+1\right)>>1, & otherwise\end{cases}, \tag{5.4}$$

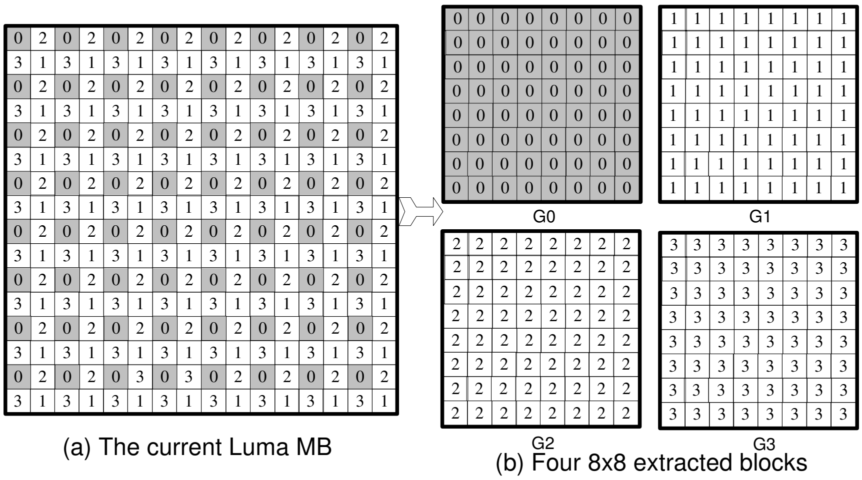
where $p_{i,j}^{m}$ denotes the predicted value of the sample in position (i,j) of group m (Gm), and $f_{i,j}^{m}$ denotes the reconstructed value of the reference sample in the position (i,j) of Gm.

Since the best predicted direction (horizontal or vertical/diagonal down left or diagonal down right) of each sample in the current block can be determined adaptively based on the measure of intensity gradient according to the neighboring reconstructed samples, there is no extra bit required to encode the predicted direction. Moreover, since samples in G1, G2 and G3 are predicted using the neighboring reconstructed samples which are equal to the original samples in a lossless coding, the correlation between them is generally higher. As a result, the information left in the residual can be reduced significantly compared with the original block-based method.

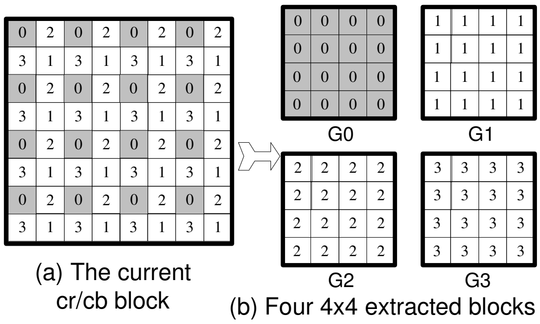
5.2.2 HIP method for I16MB type /Chroma prediction.

Similar to I8MB type (except the block size), the whole MB with a 16x16 size is classified into four groups from G0 to G3 as shown in Figure 5.4(a). Samples in G0 are predicted using four modes in I16MB type based on the samplewise DPCM method. Also two coding schemes are adopted to optimize the residual encoding for each 8x8 block. The coding scheme with the smaller rate cost is selected as the encoding method of the residual block produced by one of four prediction modes. The mode with the minimal rate cost is selected and encoded. Samples in G1, G2 and G3 are then predicted using the neighboring reconstructed samples as (5.2)-(5.4). The prediction of Chroma cr/cb block can also be performed in a similar way to the

HIP method of the I16MB type. The structure of HIP for a Chroma block is shown in Figure 5.4(b).



(a)



(b)

Figure 5.4. Structure of HIP: (a) for I16MB type, (b) for Chroma prediction

Based on the above analysis, we can see that, for each MB, only part of samples are predicted using the rate optimization (RO) process based on the proposed algorithm. As a result, the complexity of the encoder cannot be increased dramatically in spite of two residual coding schemes. Another observation is that the best prediction directions of the samples in the last three groups can be select adaptively based on the minimal gradients. Hence, the prediction directions of the samples in a block may be different using the proposed algorithm. This cannot be obtained using the previous intra prediction methods [5, 26-27], and it is also one of the reasons that the proposed algorithm can show a better compression performance. This last point is to be verified in Section 5.2.5.

5.2.3 Further improvement of HIP method

Compared with the I8MB type and the I16MB type, the I4MB type is associated with a 4x4 block, which is small. Experimental results show that the hierarchical intra prediction method cannot have a better effect on I4MB type. Therefore, only the samplewise DPCM method in [26] is used to improve the prediction of I4MB type for lossless coding. Note that the samplewise DPCM concept was just applied to four modes (modes 0, 1, 3 and 4) in [26]. The performance is thus limited. In [27], the authors extended the samplewise DPCM concept to other intra prediction modes by using a simple interpolation method. As a result, further improvement is achieved. Note that the sample DPCM method shows better prediction accuracy for the directional intra prediction modes in I4MB type. Therefore, we adopt the similar prediction strategy in [27] to make intra prediction for the directional modes in I4MB type. In [26], the authors said that the samplewise DPCM method is not suitable to the DC mode since more than one reference samples are used to perform interpolation. However, extensive experimental results show that the DC mode cannot show a better performance since the average value is used as the predicted values of the whole MB/block. Note that the DC mode is assigned a relatively small mode index (2) among all prediction modes in the H.264/AVC standard, which means that this mode has a higher probability to be selected as the best prediction mode. Therefore, improving the prediction structure of the DC mode has a great significance for an efficient lossless coding. The method as shown in (5.5) is proposed to improve the prediction of the DC mode in every intra prediction type

(I4MB type, I8MB type, I16MB type or Chroma prediction) according to the availability of the neighboring reference pixels.

$$p_{i,j}^{0} = \begin{cases} \left(f_{i,j-1}^{0} + f_{i-1,j}^{0} + 1\right) >> 1, & up\_available \\ \left(f_{i-1,j}^{0} + f_{i-1,j+1}^{0} + 1\right) >> 1, & left\_available \\ 128, & otherwise \end{cases}, \tag{5.5}$$

where coordinates (i,j) correspond to samples in G0. Similarly, in order to use the concept of DPCM for the plane mode in the I16MB type and the Chroma prediction, the predicted value of the sample in G0 for the plane mode is given as follows:

$$p_{i,j}^{0} = \left(f_{i,j-1}^{0} + f_{i+1,j}^{0} + 1\right) >> 1 \tag{5.6}$$

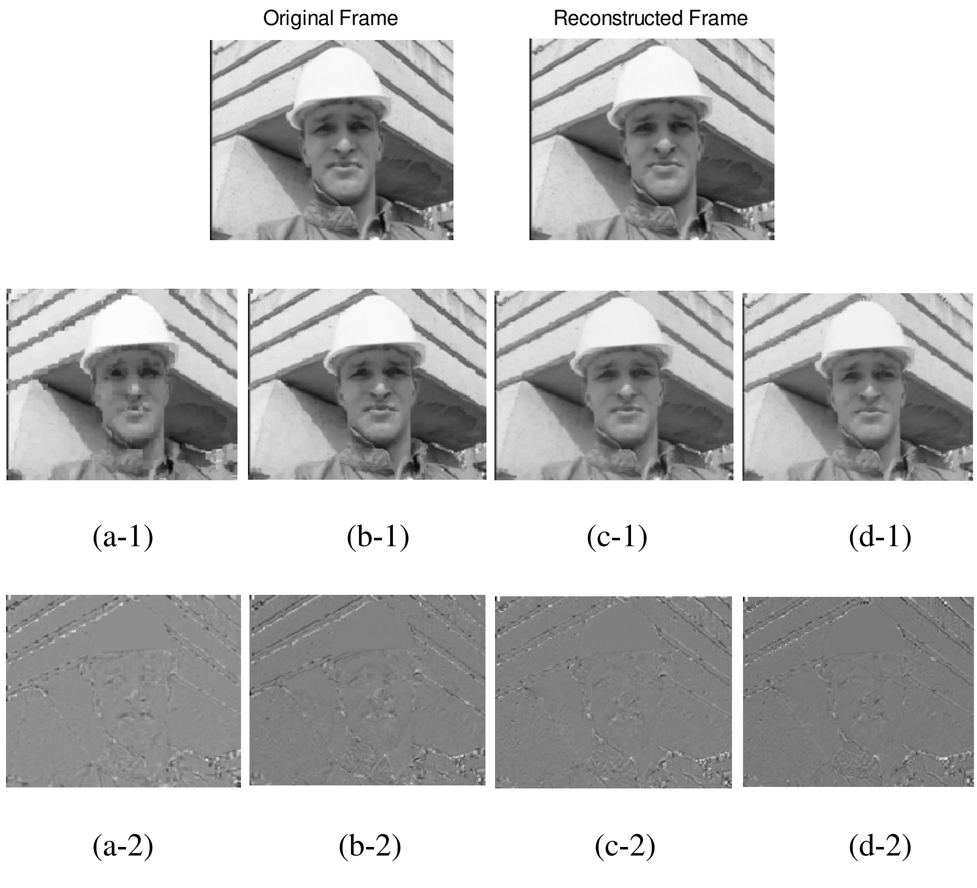


Figure 5.5. Experimental results on Luma component for the first frame of Foreman.QCIF sequence: (a-1/2) Predicted/Residual frame based on H.264/AVC, (b-1/2) Predicted/Residual frame based on DPCM, (c-1/2) Predicted/Residual frame based SI_DPCM, (d-1/2) Predicted/Residual frame based on HIP

In addition, two coding schemes are also applied to the residual blocks produced by the nine prediction modes of the I4MB type in order to efficiently compress the residual. Figure 5.5 shows an example of an original frame, reconstructed frame, predicted frame and their corresponding residual frames for several lossless coding algorithms. From the results, it can be observed that the reconstructed frame is the same as original frame since it is a lossless coding process. Another observation from the predicted frames is that the proposed algorithm makes a best prediction among all these algorithms, and thereby the information left in the residual frame (d-2) is the smallest.

5.2.4 Encoding of resultant residual blocks

5.2.4.1 Formation of two coding modes

Note that lossless compression techniques usually include two major components: prediction and entropy coding. Having a better prediction method can result in less information in the residual block, which can be illustrated by Figure 5.5. Also a proper entropy coding is equally important for compressing the residual block. It is well-known that the CAVLC method [1-4] is mainly designed to compress the QDCT coefficients, so it is not efficient to compress the residual block directly if the distribution of the residual is different from that of the QDCT coefficients of the residual block. Therefore, the CAVLC method should be improved correspondingly for a lossless coding to obtain better compression efficiency. Actually, an improved CAVLC method for lossless residual coding has been investigated in [67]. However, this method is proposed based on the statistics of the residual produced by the lossless coding method in the H.264/AVC standard. In other words, the method in [67] is highly dependent on the distribution of the residual. Different residual distributions require different coding design. As a result, it is hard to apply this method directly to the residual produced by other lossless coding methods. Different from improving CAVLC structure, we propose a general method to improve the compression efficiency from the view of residual block in lossless coding.

After some investigation, we find that some residuals show similar characteristics to that of the QDCT coefficients and others do not. According to these two cases, we propose two coding schemes to compress the residual block, which produce efficient results as shown below. Figure 5.6 shows the block diagram about the coding scheme selection process.

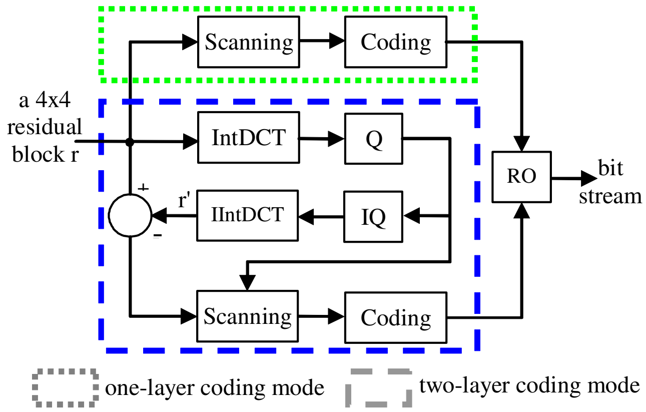


Figure 5.6. Coding scheme selection based on the rate optimization

As shown in Figure 5.6, there are two coding schemes: one-layer coding mode and two-layer coding mode, designed for encoding the residual block. In the one-layer coding mode, the residual block is directly scanned and entropy coded by entropy method. While the two-layer coding mode includes a lossy coding process and a lossless coding process. The residual block is firstly transformed by integer DCT (IntDCT) and quantized to obtain the QDCT coefficients in the lossy coding process. The QDCT coefficients are scanned and entropy coded. While for the lossless coding process, inverse quantization (IQ) and inverse integer DCT (IIntDCT) are applied to the QDCT coefficients to obtain the reconstructed residual block r'. The difference between the original residual block r and the reconstructed residual block r' are scanned and entropy coded.

5.2.4.2 Selection criterion of the better coding mode

How to select a better coding scheme from the two candidate schemes is a critical problem during the encoding. In the reference software JM of the H.264/AVC standard, the Lagrangian optimization is introduced to make prediction mode decision. It aims at minimizing the distortion under a constraint on the rate. However, there is no distortion for a lossless coding. Even though the distortion can be found in the lossy coding process of the two-layer coding scheme, it does not require to be optimized since the compression ratio is the only element that we care about in a lossless coding. As a result, a rate optimization method is suggested by us to select a better residual coding scheme. The rates corresponding to the two coding schemes can be obtained by (5.7) and (5.8), respectively. The coding scheme with a smaller rate cost is selected for each residual block. By using the rate optimization method, the largest compression efficiency can be obtained.

$$J_{OLC} = R(residual \mid \mathrm{mod}\, e) \tag{5.7}$$

$$J_{TLC} = R_{lossy}(QDCT \mid \mathrm{mod}\, e, QP) + R_{lossless}(distortion \mid \mathrm{mod}\, e, QP), \tag{5.8}$$

where QP is a quantization parameter. J_{OLC}(R(residual|MODE)) and J_{TLC} represent the number of bits used to encode the residual block by the one-layer coding scheme or the two-layer coding scheme, respectively. R_{lossy}(QDCT|MODE,QP) represents the number of bits used to encode the QDCT coefficients, and $R_{lossless}$(distortion|MODE,QP) represents the number of bits used to encode the distortion between r and r'. Note that "MODE" is used to represent one of the nine intra prediction modes for G0 block which requires a mode decision process. For other blocks, such as G1, G2 and G3, (5.9) and (5.10) should be changed into (5.9) and (5.10), respectively. Because these blocks are predicted adaptively based on a gradient measure method without mode decision process.

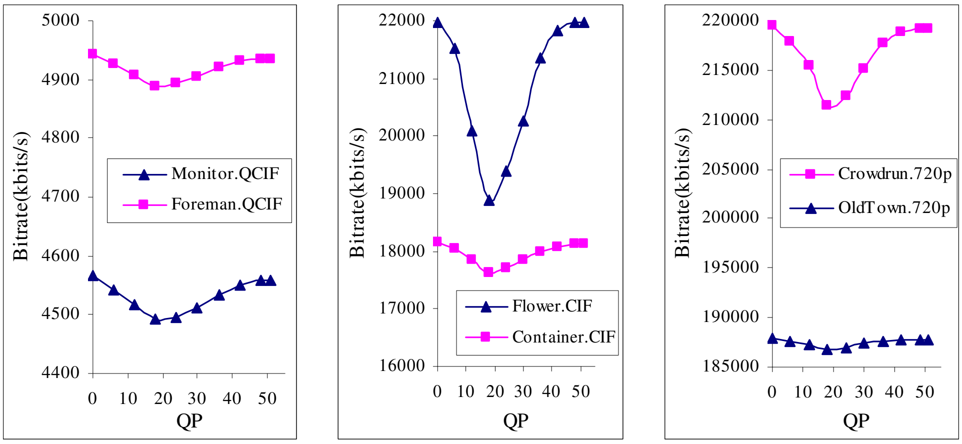
$$J_{OLC} = R(residual) \tag{5.9}$$

$$J_{TLC} = R_{lossy}(QDCT \mid QP) + R_{lossless}(distortion \mid QP) \tag{5.10}$$

Basically, when the residual block has similar characteristics with the QDCT coefficients, the one-layer coding scheme is usually the better selection. By contrast, if the characteristics of the residual are very different from the QDCT coefficients, the two-layer coding scheme shows a better compression performance. Compared with the method in [28], where only a two-layer coding method is used, our method can provide the better compression ratio by selecting a better one from two coding schemes based on the rate optimization method. Another improvement compared with the method in [28] is that the rates in both lossy layer and lossless layer for the two-layer coding scheme are combined together to be optimized for both prediction mode decision and coding scheme selection. This makes the proposed algorithm give the best compression efficiency, which will be verified experimentally in Section 5.2.5.

5.2.4.3 Setting of QP for the two-layer coding mode

As we have mentioned, the two-layer coding scheme associates with a quantization process. Therefore, a quantization parameter must be specified during encoding process. In order to achieve the best compression efficiency for lossless coding, more than 20 video sequences were encoded with QP from 0 to 51 in our testing. Figure 5.7 shows an example about the relationship between bitrate and QP for six video sequences with different frame sizes. According to the experimental results, QP is set to 18 for achieving the best compression efficiency.



(a) (b) (c)

Figure 5.7. Relationship between bitrate and QP

5.2.4.4 Encoding of the flags to indicate the resultant coding schemes in an MB

Due to the introduction of two coding schemes, we have to define a flag (0: one-layer coding scheme; 1: two-layer coding scheme) to indicate which scheme is used during the encoding process, and this flag will also be written into the bit stream. Figure 5.8 shows an example about the flag map of one MB encoded by I4MB or I8MB.

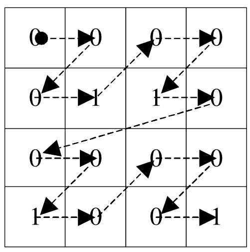


Figure 5.8. A map of flags for coding schemes of 16 4x4 blocks

To maximally increase entropy coding efficiency, we propose an improved context-based adaptive variable length coding scheme (ICAVLC) to encode the flags.

Note that these flags only contain “ones” and “zeros”. Based on this observation, three syntax elements: *total_zeros*, *run_before* and *all_zero_flag* are used to efficiently encode these flags in ICAVLC. Similar to the syntax elements used in CAVLC of H.264/AVC standard, the "*total_zeros*" represents the total number of zeros before the last "1" (in reverse order), and "*run_before*" is used to indicate the number of consecutive zeros between two "1"s. The "*all_zero_flag*" is designed to indicate whether these 16 flags in an MB are all zero values. "*all_zero_flag* = 1" denotes that the sixteen flags include only zero values, and "*all_zero_flag* = 0" means that at least one "1" is found in these 16 flags. The proposed ICAVLC method can be realized by the following steps:

Step 1: Reorder the flags in an MB according to the scanning method as shown in Figure 5.8 to produce a linear array: 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1.

Step 2: Encode *all_zero_flag*. If *all_zero_flag* = 1, terminate the encoding process. Otherwise go to Step 3.

Step 3: Encode "*total_zeros*" using four bits.

Step 4: Encode "*run_before*" in reverse order according to Table 9-10 in [4]. Note that the last *run_before* is also encoded into the bit stream since there is no syntax element used to represent the number of flag "1". In order to resolve this problem, we add a binary codeword "000000000001" to the bottom of Table 9-10 in [4] to represent the case when the *run_before* is 15.

Different from the HIP methods of I4MB and I8MB types, where 16 flags are needed for an MB, only four flags corresponding to four 8x8 blocks as shown in Figure 5.4(a) are required for an MB based on the HIP method of I16MB type. In this case, we use four bits to represent the coding schemes of the four 8x8 blocks.

Table 5.1 lists the percentage of the bits used for coding the flags in total bits of each sequence. From this table, we can see that the percentage of the bits used for coding the flags only accounts for 0.126% of the whole bits on average. This is a very small proportion. Although the bits are increased due to coding the flags, the proposed algorithm can compensate for these bits, and provide a higher compression efficiency than other algorithms in the literature as shown in Tables 5.2-5.4. Table 5.1 also shows the saving of bits used in the flags for I4MB and I8MB types, compared with the coding method of "one flag one bit". On average, 25.342% saving can be achieved based on the table.

Table 5.1. Statistical results for coding schemes of 16 4x4 blocks

Seq.		**Saving (%)**	**Percentage in Total Bits (%)**
QCIF	Coastguard	14.023	0.124
	Monitor	23.047	0.129
	Foreman	13.243	0.139
	Container	30.960	0.151
CIF	Children	13.497	0.143
	Flower	51.898	0.064
	Hall	21.158	0.141
	Akiyo	37.265	0.138
HDTV (720p)	Sunflower	27.057	0.157
	Crowdrun	9.740	0.113
	Shuttlestart	32.740	0.102
	OldTownCross	29.476	0.108
Average		**25.342**	**0.126**

5.2.5 Experimental Results

In order to evaluate the performance of the proposed algorithm, the reference software JM 12.2 [5] was used to carry out experiments on a number of YUV 4:2:0 format sequences. The test was based on the FRExt high profiles with specifications provided in [59]. All frames were intra coded. Some video sequences referring to

Tables 5.2-5.4 were used for this work. A total of 150 frames were encoded for each sequence. The frame rate was set to 30. CAVLC entropy coding method was used for the experiments. Deblocking was not used for lossless coding.

Since there is no degradation in video quality for lossless coding, we only present the compression ratios for different lossless intra prediction methods in terms of encoding bitrates for sequences of different sizes in Tables 5.2-5.4. It can be observed from these tables, the method based on samplewise DPCM [26] gives a higher compression ratio than that of the block-based method in the original H.264/AVC standard. With the improvements of more intra prediction modes, the method based on SI_DPCM [27] achieves a further compression ratio compared with the samplewise DPCM method. Compare with samplewise DPCM method, the proposed method can achieve a consistent increase in compression ratios for all test sequences. For most of sequences, the proposed method also shows a better compression ratio compared to that of the SI_DPCM method. Therefore, we can see that the proposed algorithm can show the best compression efficiency compared with the methods [5, 26-27] based on the H.264/AVC framework. Note that only part of samples based on the proposed algorithm are predicted using the multiple prediction modes defined in the H.264/AVC standard. As a result, the complexity of the encoder using the proposed algorithm is still less than the method in [28]. Furthermore, the proposed algorithm may be used together with the inter-frame prediction approach used in H.264/AVC standard, and thereby the lossless compression efficiency of the encoder can also be greatly improved. Note that some fast algorithms such as [52, 60, 61, 68] can also be combined with this algorithm, therefore, it gives a possible better selection for real-time applications.

Table 5.2. Results of compression ratios for QCIF sequences

Sequence (QCIF)	**Bitrate (kbits/s)**	**Δ Bitrate [%]**		
	H.264_LS	H.264 _DPCM	SI_ DPCM	Proposed method
Coastguard	5855.984	10.574	11.846	14.983
Monitor	5377.597	11.853	14.863	16.547
Foreman	5566.776	9.026	11.123	12.274
Akiyo	4659.207	10.843	15.904	17.767
Mobile	8082.437	7.287	10.363	17.698
Carphone	4945.672	11.237	14.510	15.267
Container	5230.263	10.185	12.507	15.702
TableTennis	5669.637	8.190	11.269	13.588
Salesman	5966.341	11.692	15.636	16.573
GrandMother	5306.482	9.132	11.572	12.737
Average	**5666.040**	**10.002**	**12.960**	**15.314**

Table 5.3. Results of compression ratios for CIF sequences

Sequence (CIF)	**Bitrate (kbits/s)**	**Δ Bitrate [%]**		
	H.264_LS	H.264 _DPCM	SI_ DPCM	Proposed method
Children	24264.276	15.280	23.389	22.668
flower	23235.798	3.648	5.754	18.784
Container	20372.872	8.152	10.711	13.571
Hall	20037.900	11.450	13.913	14.838
Akiyo	16562.680	8.763	13.217	13.305
Foreman	19353.068	8.083	10.410	11.352
Coastguard	23328.964	10.615	12.444	14.876
Silent	21750.124	8.450	12.036	12.321
Tempete	26259.104	9.532	13.135	18.273
Waterfall	27073.632	7.946	12.964	14.895
Average	**22223.840**	**9.192**	**12.797**	**15.488**

Table 5.4. Results of compression ratios for HDTV sequences

Sequence (HDTV)	**Bitrate (kbits/s)**	**Δ Bitrate [%]**		
	H.264_LS	H.264 _DPCM	SI_ DPCM	**Proposed method**
Sunflower	168292.240	15.361	22.372	20.707
Crowdrun	241153.440	5.462	8.035	12.429
InToTree	198065.424	2.210	2.809	4.491
OldTownCross	199703.840	4.528	4.661	6.539
Average	**201803.7**	**6.890**	**9.469**	**11.042**

5.3 Lossless intra coding algorithm based on adaptive interpolation filter and residual coding optimization in H.264/AVC

Note that the number of prediction directions in the subblock is still finite by using the first algorithm, which affects accuracy of the prediction. It is well know that adaptive interpolation techniques [69-72] have the ability to tune the interpolation coefficients to match an arbitrarily-oriented step edge. Note also that a filter with adaptive coefficients is proposed for the interpolation of MB in [73, 74] for the high efficiency video coding [10, 75]. The adaptive interpolation filter has achieved a good performance for fractional-pixel ME [76-78]. After observing this advantage of adaptive interpolation filter in possibly formulating a high-resolution reference frame for high efficiency video coding, we propose a new lossless intra prediction algorithm with the emphasis on interpolation techniques.

5.3.1 Hierarchical Block Pattern

In order to improve the prediction accuracy of samples away from their references, we suggest that pixels in an MB/block also be predicted hierarchically as shown in Figures 5.1 and 5.4 for the I8MB type, I16MB type and the Chroma component. As shown in the two figures, the pixels in each prediction unit specified in H.264/AVC, such as 8x8 for I8MB type, 16x16 for I16MB type and 8x8 for Chroma component are sampled into four subblocks, which are labeled as from G0 to G3. Note that samples labeled as 0 in G0 can be seen as a low-resolution block, which comes from a decimation/down-sample result of the original 8x8/16x16 block. In other words, the recovery/prediction of the original 8x8/16x16 block thus becomes an inverse problem. Therefore, we refer to the samples labeled by 0 as reference, and aim at constructing the 8x8/16x16 block related to G0 as its super resolution version. Details are given as follows.

5.3.2 Hierarchical Intra Prediction

In order to have the super revolution version, samples in G0 should be firstly predicted, encoded and reconstructed in the encoder side. Samples in G1 are then predicted using the samples in G0 as references. Subsequently, samples in G2 and G3 are predicted using samples in G0 and G1 as references. Details about the proposed intra prediction algorithm are shown as follows.

5.3.2.1 Intra prediction for samples in G0 based on the standard method

The samples in G0 are firstly predicted using intra spatial prediction (ISP) in H.264/AVC. In fact, other intra prediction algorithms can be applicable here based on the encoders to be used. To improve the predicted accuracy, the DPCM concept in [26] is extended to each mode in the intra prediction of H.264/AVC. Note that only the samples labeled as 0 are predicted as shown in Figure 5.9. Taking the vertical prediction in I8MB type as an example, the predicted values (p_{ij}) are given as follows by referring to Figure 5.9(a).

Vertical prediction:
$p_{00}=f_a$, $p_{02}=f_{00}$, $p_{04}=f_{02}$, $p_{06}=f_{04}$;
$p_{20}=f_c$, $p_{22}=f_{20}$, $p_{24}=f_{22}$, $p_{26}=f_{24}$;
$p_{40}=f_e$, $p_{42}=f_{40}$, $p_{44}=f_{42}$, $p_{46}=f_{44}$;
$p_{60}=f_h$, $p_{62}=f_{60}$, $p_{64}=f_{62}$, $p_{66}=f_{64}$;

where f_{ij} denotes the pixel value in position (i, j) of the current block.

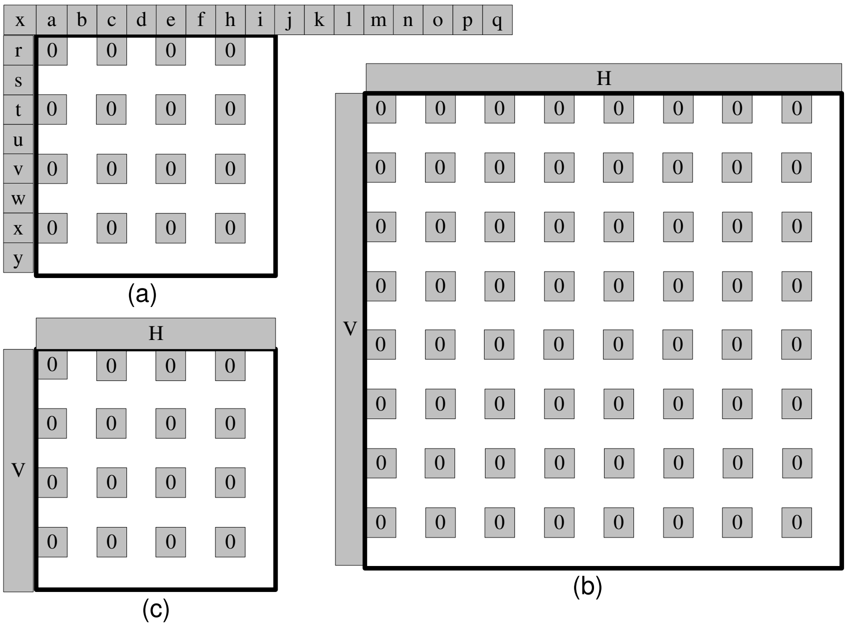


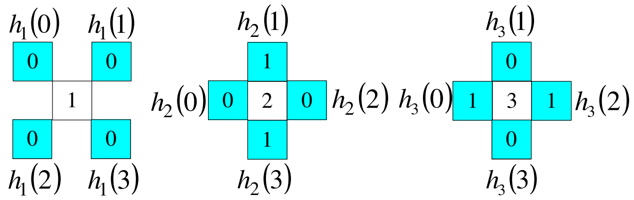
Figure 5.9. Illustration of samples to be predicted in G0

(a) I8MB, (b) I16MB and (c) Chroma prediction

After the prediction is finished, the difference between the original block and the predicted block, i.e., the residual block, is encoded to form a bitstream. On the other hand, G0 can be reconstructed in the encoder side. Since it is a lossless coding process, the reconstructed G0 is the same as the original G0.

5.3.2.2 Intra prediction for samples in G1, G2 and G3 based on adaptive interpolation filters

To construct the 8x8/16x16 block, namely, for predicting the samples labeled as 1, 2 and 3, a non-separable 2D adaptive interpolation filter (AIF) with the fourth order is applied. The predicted structures for pixels in different subblocks are specified as shown in Figure 5.10.



(a) (b) (c)

Figure 5.10. Predicted structures of samples in (a) G1, (b) G2 and (c) G3

Taking Figure 5.10 (a) as an example to illustrate the prediction process, the model for samples in G1 can be described as

$$p_{i,j}^{1} = \sum_{m=0}^{1}\sum_{n=0}^{1} h_1(2m+n) f_{i+m,j+n}^{0} , \tag{5.11}$$

where i, j=0,1,2,3 respectively (same as below). Symbol $p_{i,j}^{1}$ denotes the predicted value of the sample in position (i, j) of the extracted block G1, and $f_{x,y}^{0}$ denotes the pixel value in position (x, y) of the extracted block G0. In our implementation, the filter length is set to 4 for sake of simplicity. In fact, it can be any other meaningful values. The four filter coefficients {h_1(k), for k=0, 1, 2, 3} are calculated for the samples in G1 by the minimisation of the prediction error energy. By using the obtained filter coefficients and the reference pixels in G0, the predicted value of each pixel in G1 is calculated according to (5.11).

In the process of minimization, the error of the prediction is calculated as

$$e_{i,j}^{1} = f_{i,j}^{1} - p_{i,j}^{1} \tag{5.12}$$

Note that the desired sample $f_{i,j}^{1}$ is known in the encoder side but not available in the decoder side. In order to have a decodable system, one feasible way is to encode these filter coefficients into the bitstream. However, four filter coefficients are required to be encoded for each 4x4/8x8 block. As a result, it may increase the total bits for a sequence. Therefore, to encode these filter coefficients is not a good choice. Samples in G0 (as a low-resolution subblock) are available when we predict samples in G1 (as a high-resolution subblock). Note that a natural image source can always be modeled as a locally stationary Gaussian process [79]. Hence the filter coefficients can be estimated by using the geometric duality between the low-resolution covariance and the high-resolution covariance. Geometric duality is the correspondence between the high-resolution covariance and the low-resolution covariance that connects the pixels along the same orientation but with different resolutions [79, 80]. Figure 5.11 shows the geometric duality between the high-resolution covariance and the low-resolution covariance for samples in the three extracted blocks.

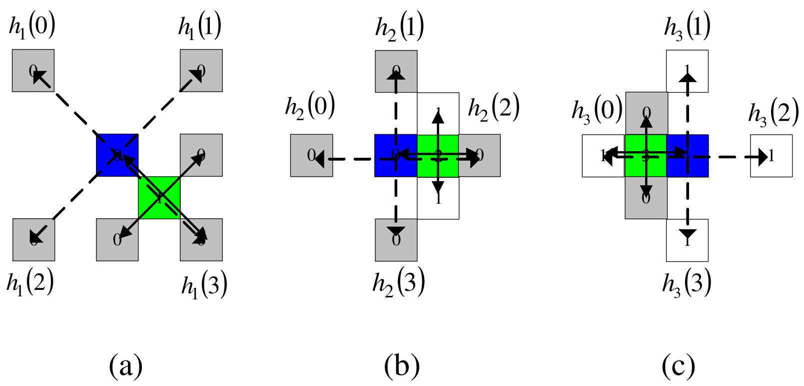


Figure 5.11. Geometric duality for the interpolation of samples in (a) G1, (b) G2 and (c) G3

For example, in Figure 5.11 (a), the sample in blue color should be predicted instead of predicting the sample in green color based on geometric duality. The predicted error is given as follows:

$$e_{i,j}^{0} = f_{i,j}^{0} - p_{i,j}^{0} \tag{5.13}$$

$$p_{i,j}^{0} = \sum_{m=0}^{1}\sum_{n=0}^{1} h_1(2m+n) f_{2(i+m),2(j+n)}^{0} \tag{5.14}$$

For each 4x4/8x8 block, four filter coefficients have to be solved by minimizing the mean-square error of the original pixel value $f_{i,j}^{0}$ and the predicted value $p_{i,j}^{0}$. After mathematical manipulation, the filter coefficients in matrix form ($\mathbf{H}^{\text{opt}}$) can be represented as follows:

$$\mathbf{H}_c^{opt} = \left(\mathbf{R}_{ff}^{c}\right)^{-1} \mathbf{R}_{fd}^{c} \tag{5.15}$$

In (5.15), $\mathbf{H}^{\text{opt}}$, $\mathbf{R}_{ff}^{c}$ and $\mathbf{R}_{\text{fd}}$ are expressed as follows:

$$\mathbf{H}_c^{opt} = \left[h_c^{opt}(0) \quad h_c^{opt}(1) \quad h_c^{opt}(2) \quad h_c^{opt}(3)\right]^T \tag{5.16}$$

$$\mathbf{R}_{ff}^{c} = \frac{1}{N^2}\mathbf{C}^T\mathbf{C} \tag{5.17}$$

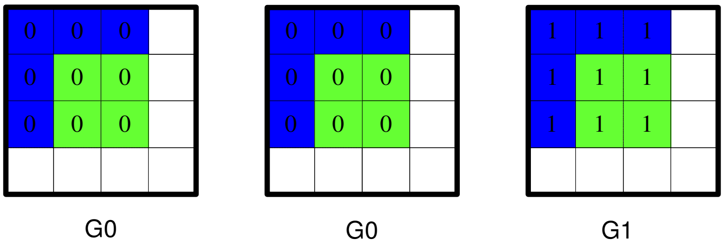
$$\mathbf{R}_{fd}^{c} = \frac{1}{N^2}\mathbf{C}^T\mathbf{F}, \tag{5.18}$$

where N^2 represents the number of pixels used in the training window. **F** and **C** are denoted as follows:

$$\mathbf{F} = \begin{bmatrix} f_{0,0}^{c} \\ f_{1,0}^{c} \\ \vdots \\ f_{N-1,N-1}^{c} \end{bmatrix}, \text{ and } \mathbf{C} = \begin{bmatrix} s_{10}^{c} & s_{11}^{c} & s_{12}^{c} & s_{13}^{c} \\ s_{20}^{c} & s_{21}^{c} & s_{22}^{c} & s_{23}^{c} \\ \vdots & \vdots & \vdots & \vdots \\ s_{N^2 0}^{c} & s_{N^2 1}^{c} & s_{N^2 2}^{c} & s_{N^2 3}^{c} \end{bmatrix}$$

In fact, **F** is a vector composed of the training data with NxN pixels inside the training subblock, **F** and **C** is an N^2x4 matrix whose i$^{\text{th}}$ row vector consists of four predictors { s_{ij}^{c}, for j=0, 1, 2, 3} as shown in Figure 5.11 for predicting the i$^{\text{th}}$ element in **F**. Similarity, the four filter coefficients {$h_2(m)$ and $h_3(m)$, for m=0, 1, 2, 3} are calculated for the pixels in G2 and G3 based on geometric duality, respectively. The predicted value of each pixel in G2 or G3 is then calculated by using these filter coefficients and the reference pixels in G0 and G1 by referring to Figure 5.10. The residual block is then encoded into bitstream.

Theoretically, the size of the training window should be equal to the size of subblock G1, G2 and G3 since each predicted sample in G1, G2 and G3 has a correspondence based on the geometric duality. However, some samples located at the boundary of an MB/block as shown in Figures 5.1 and 5.4 cannot have four predictors. As a result, the training window of an adaptive interpolation filter for G1, G2 or G3 is indicated in dark color (green and blue colors) as shown in Figure 5.12.



(a) (b) (c)

Figure 5.12. Training window of Adaptive Interpolation for (a)G1, (b)G2 and (c)G3

Note that the training samples labeled in dark color (blue color) do not always have four predictors. When there are less than four predictors, the samples in G1, G2 and G3 are predicted by using a simple intensity gradient measurement method instead of using the proposed AIF method. In the intensity gradient measurement, the predicted value of each sample in G1, G2 and G3 is given by (5.19), (5.20) and (5.21), respectively.

$$p_{i,j}^{1} = \begin{cases} \left(f_{i-1,j-1}^{0} + f_{i+1,j+1}^{0} + 1\right) >> 1, & \left|f_{i-1,j-1}^{0} - f_{i+1,j+1}^{0}\right| < \left|f_{i+1,j-1}^{0} - f_{i-1,j+1}^{0}\right| \\ \left(f_{i+1,j-1}^{0} + f_{i-1,j+1}^{0} + 1\right) >> 1, & otherwise \end{cases} \tag{5.19}$$

$$p_{i,j}^{2} = \begin{cases} \left(f_{i-1,j}^{0} + f_{i+1,j}^{0} + 1\right) >> 1, & \left|f_{i-1,j}^{0} - f_{i+1,j}^{0}\right| < \left|f_{i,j-1}^{1} - f_{i,j+1}^{1}\right| \\ \left(f_{i,j-1}^{1} + f_{i,j+1}^{1} + 1\right) >> 1, & otherwise \end{cases} \tag{5.20}$$

$$p_{i,j}^{3} = \begin{cases} \left(f_{i-1,j}^{1} + f_{i+1,j}^{1} + 1\right) >> 1, & \left|f_{i-1,j}^{1} - f_{i+1,j}^{1}\right| < \left|f_{i,j-1}^{0} - f_{i,j+1}^{0}\right| \\ \left(f_{i,j-1}^{0} + f_{i,j+1}^{0} + 1\right) >> 1, & otherwise \end{cases} \tag{5.21}$$

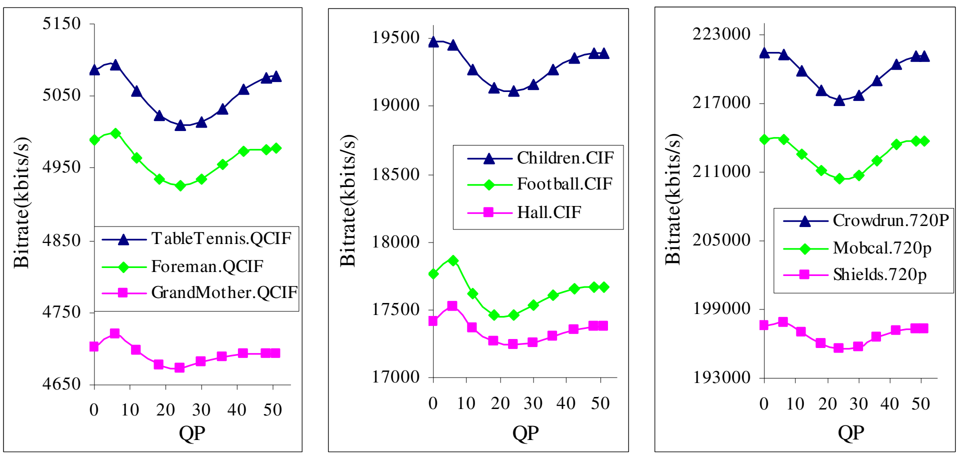
Since samples in G1, G2 and G3 are predicted using the neighboring reconstructed samples which are equal to the original samples in a lossless coding process, the correlation between them is generally high. Moreover, the adaptive interpolation filter is used to predict the samples in G1, G2 and G3. It can spatially adapt the filter coefficients to the local structures. As a result, samples in G1, G2 and G3 can be well predicted. Note that the adaptive interpolation filter is performed on a block of pixels (G1, G2 or G3) rather than one by one. As a result, the computational load will not be highly increased.

5.3.2.3 Intra prediction for I4MB type

Compared with the I8MB type, the I16MB type and the Chroma prediction, the I4MB type is associated with a small block. Experimental results show that the hierarchical intra prediction method does not have a better effect on I4MB type. Therefore, only the samplewise DPCM method is used to improve the prediction of I4MB type for lossless coding.

5.3.3 Encoding of resultant residual blocks

By extending the adaptive interpolation filter to the hierarchical intra prediction, a more accurate prediction method is generated. Thus, less information is left in the residual block. To have an efficient compression to cater for the different distributions of residual samples, two coding modes: one-layer coding (OLC) mode and two-layer coding (TLC) mode presented in Section 5.2.4, are adopted to compress a 4x4 residual block. Similarly, in the one-layer coding mode, the residual block is directly scanned and entropy coded. While the two-layer coding mode includes a lossy coding process and a lossless coding process. In the lossy coding process, the residual block is firstly transformed by integer DCT and quantized to obtain the QDCT coefficients. The QDCT coefficients are scanned and entropy coded. While for the lossless coding process, inverse quantization and inverse integer DCT are applied to the QDCT coefficients to obtain the reconstructed residual block r'. The differences between the original residual block r and the reconstructed residual block r' are scanned and entropy coded. Figure 5.6 shows the block diagram about the optimization process of the two coding modes. For the two coding modes, the rate optimization method presented in Section 5.2.4 is used to select the better one. Note that a quantization process as shown in Figure 5.6 is involved in the two-layer coding mode. Therefore, a quantization parameter (QP) must be specified during the encoding process. Due to a different method adopted to the prediction of G1, G2 and G3, the QP should be re-specified for an efficient compression. In our experimental work, more than 30 video sequences were encoded with QP from 0 to 51.



(a) (b) (c)

Figure 5.13. Relationship between bitrate and QP

Figure 5.13 gives an example about the relationship between the bitrate and QP for nine typical tested sequences with different sizes (QCIF, CIF or HDTV). For every three sequences in (a), (b) or (c) of Figure 5.13 have different texture details and motion activities. The sequences with the higher bitrate in (a), (b) or (c) usually have complex textures and motion activities. According to the experimental results, our proposed algorithm consistently achieves the best compression efficiency for all kinds of sequences, when the QP is set to 24 for the two-layer coding mode.

Since two candidate coding modes are adopted for each 4x4 residual block, a flag (0: one-layer coding mode; 1: two-layer coding mode) is required to indicate which mode is better during the encoding process, and this flag needs to be written into the bit stream. The ICAVLC method proposed in Section 5.2.4 is used to encode the flags in an MB for I4MB and I8MB types. Due to only four flags corresponding to four 8x8 blocks in an MB as shown in Figure 5.4 (a) required in I16MB type, four bits are used to represent the four coding modes in an MB.

5.3.4 Flowchart of proposed lossless intra compression algorithm

To have a comprehensive description about our proposed algorithm, Figure 5.14 gives a flowchart on how an MB (including Luma and Chroma components) is encoded into the bit stream by using the proposed algorithm when the lossless coding is turned on for an encoder based on the H.264/AVC standard. Given an MB, the Luma and Chroma components are encoded separately. For the Luma component, I4MB type is firstly checked. It includes intra spatial prediction (ISP) with DPCM method and coding mode selection (CMS). The I8MB and I16MB types are subsequently checked. Different from the intra prediction method for I4MB type, four subblocks (G0, G1, G2 and G3) are extracted from the Luma component (see Figures 5.1 and 5.4 (a)) in I8MB and I16MB types. The intra prediction and coding processes of G0 are similar to those for I4MB type. After the encoding of G0 is finished, G1, G2 and G3 are subsequently predicted based on the AIF method. Finally, rate optimization (RO) is applied to select the best type from the I4MB, I8MB and I16MB types. For the Chroma component, a similar process to I8MB and I16MB types is adopted. The only difference is that there is no CMS for Chroma residual block. Since the Chroma component can be well predicted, less information is left in the residual block. As a result, only one layer coding mode is efficient enough to encode the resultant Chroma residual block based on our experiments.

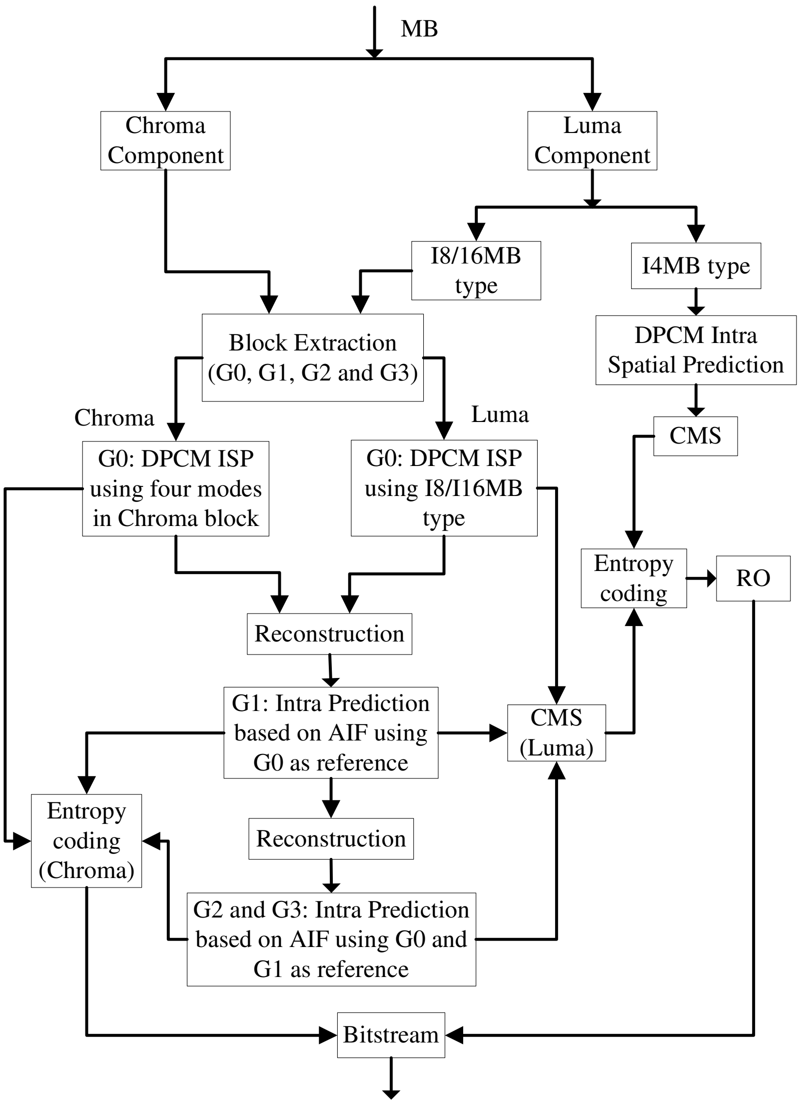


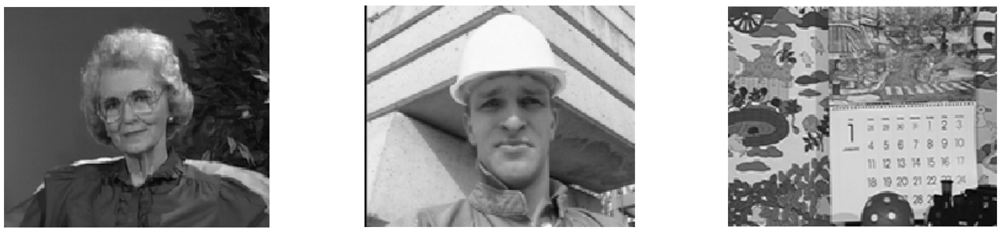
Figure 5.14. Flowchart of the proposed lossless intra compression for an MB

5.3.5 Experimental results

The proposed algorithm has been implemented on the platform of reference software JM 18.0 [5] to evaluate the performance. A large number of video sequences with YUV 4:2:0 format were used in this experiment. For each video sequence, 150 frames were encoded. The FRExt high profiles of H.264/AVC with specifications provided in [59] were set. To evaluate the efficiency of the lossless coding technique, only intra coded structure was performed. The frame rate was 30. Deblocking was disabled for lossless coding process.

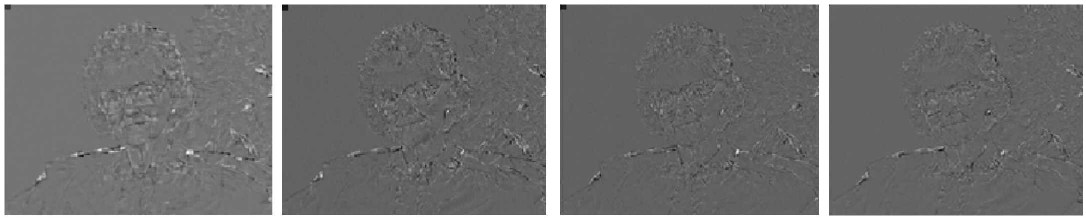
Figure 5.15 gives an example about the original frames, predicted frames and their corresponding residual frames for several lossless coding algorithms. In Figure 5.15, only the first frames of three video sequences with different texture details are shown as the representative to verify the efficiency of the proposed algorithm. It can be observed that the blocking effect in the predicted frame is obvious, such as the predicted frame in (b1)-(g1), if only block-based intra prediction is used. With the application of DPCM concept proposed in [26] for part of the intra prediction modes, the blocking effect can be reduced. For example, the predicted frames in (b2, d2 and f2) produced by using the method in [26] are more natural than the predicted frames in (b1, d1 and f1). However, the blocking effect (red circles in b2, d2 and f2 and the up-left corner) is still not eliminated. Note, for this problem, Wei et. [27] extended the DPCM concept to more numbers of intra prediction modes by using a simple interpolation (labeled as SIDPCM). The prediction becomes much better, such as that obtained in predicted frames b3, d3 and f3 in Figure 5.15. By contrast, our proposed method is a hierarchical intra prediction in combination with an adaptive interpolation filter to spatially adapt the local structures (labeled as HIPAIF). Thereby, the predicted frame is much closer to the original frame. Note also that from sequences with simple texture (such as the Grandmother in Figure 5.15) to the sequences with complex texture (such as the Mobile in Figure 5.15), the predicted frames produced by the proposed algorithm have shown a good match in quality. In other words, the proposed algorithm has shown the best prediction among all the predicted frames, and thereby the information left in the residual frames (c4, e4 and g4) is the smallest among all algorithms.

Note that the predicted values of samples in G2 and G3 are calculated purely based on the samples in G0 and G1 as shown in Figures 5.10 and 5.11. In other words, the predicted quality of G2 and G3 is dependent on the reconstruction or recovery of G0 and G1. Therefore, the residual G0 and G1 should be given a higher priority to indicate their importance, while the residual blocks G2 and G3 may be given a lower priority in poor environment if necessary. Even if the residual blocks G2 and G3 are lost during the transmission, we can still have an acceptable reconstructed frame.

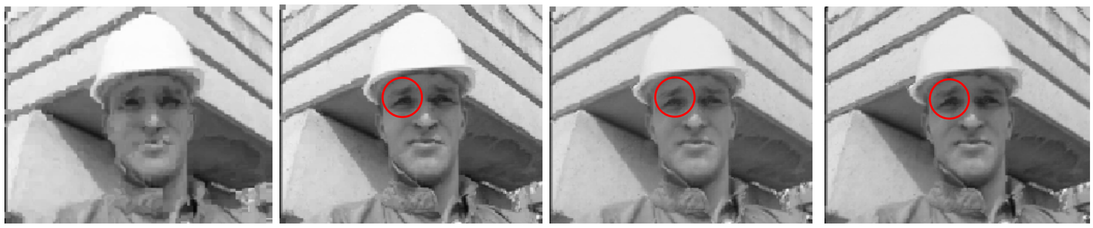


(a1) Grandmother (Original) (a2) Foreman (Original) (a3) Mobile (Original)

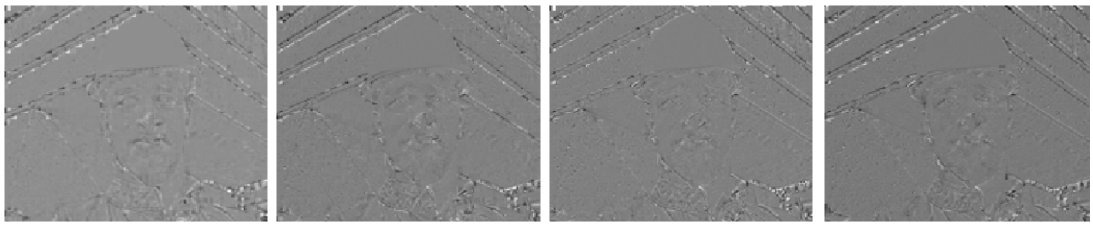
(b1) JM18.0 Predicted Frame (b2) DPCM Predicted Frame (b3) SIDPCM Predicted Frame (b4) HIPAIF Predicted Frame



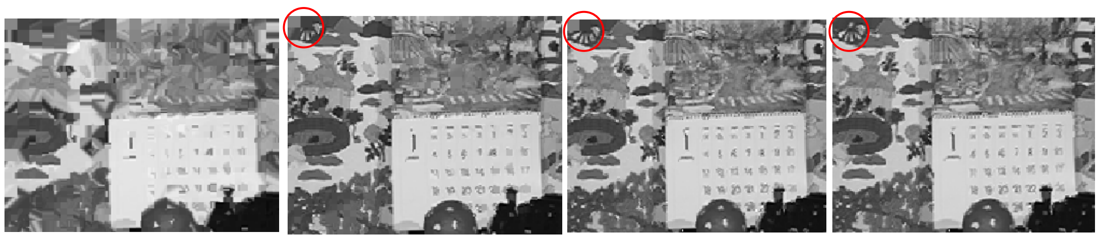
(c1) JM18.0 Residual Frame (c2) DPCM Residual Frame (c3) SIDPCM Residual Frame (c4) HIPAIF Residual Frame



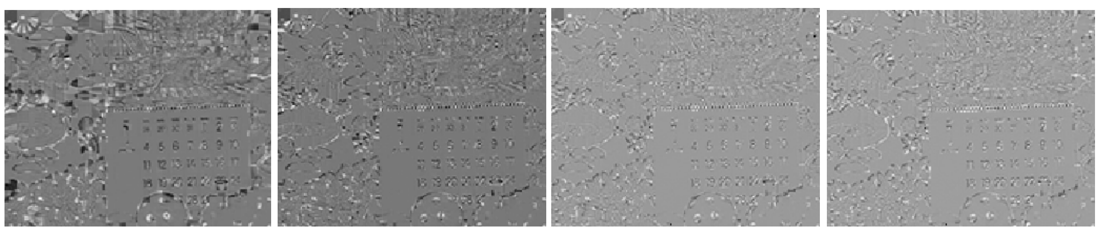
(d1) JM18.0 Predicted Frame (d2) DPCM Predicted Frame (d3) SIDPCM Predicted Frame (d4) HIPAIF Predicted Frame



(e1) JM18.0 Residual Frame (e2) DPCM Residual Frame (e3) SIDPCM Residual Frame (e4) HIPAIF Residual Frame



(f1) JM18.0 Predicted Frame (f2) DPCM Predicted Frame (f3) SIDPCM Predicted Frame (f4) HIPAIF Predicted Frame



(g1) JM18.0 Residual Frame (g2) DPCM Residual Frame (g3) SIDPCM Residual Frame (g4) HIPAIF Residual Frame

Figure 5.15. Experimental results on Luma components for the three sequences with different texture details

Table 5.5. Percentages of saving bits for QCIF sequences

Sequence (QCIF)	Bits	Saving Bits [%]			
	H.264_LS	JPEG_LS	H.264 _DPCM	SIDPCM	**HIPAIF**
Coastguard	29279040	-0.900	8.885	10.057	**12.218**
Monitor	26888775	13.550	8.330	10.568	**15.506**
Foreman	27832975	3.959	6.457	8.161	**11.502**
Akiyo	23293105	15.230	8.712	11.500	**16.755**
Mobile	40413815	13.143	3.587	4.725	**11.563**
Carphone	24726855	9.597	8.587	10.743	**14.193**
Tabletennis	28348855	6.400	5.167	6.745	**11.624**
Salesman	29832195	10.102	7.882	10.357	**15.271**
Average	**29074633**	**9.739**	**6.732**	**8.705**	**13.485**

Table 5.6. Percentages of saving bits for CIF sequences

Sequence (CIF)	Bits	Saving Bits [%]			
	H.264_LS	JPEG_LS	H.264 _DPCM	SIDPCM	**HIPAIF**
Container	101859260	5.526	5.883	7.825	**10.851**
Hall	100188750	12.089	6.517	8.654	**13.938**
Coastguard	116643040	2.720	9.193	11.055	**13.222**
Silent	108745500	5.140	4.778	6.674	**7.115**
Tempete	131293690	13.378	6.285	8.324	**11.437**
Waterfall	135369020	14.294	7.740	7.542	**14.253**
Average	**115683210**	**8.858**	**6.732**	**8.346**	**11.803**

Table 5.7. Percentages of saving bits for HDTV sequences

Sequence (720p)	Bits	Saving Bits [%]			
	H.264_LS	JPEG_LS	H.264 _DPCM	SIDPCM	**HIPAIF**
Crowdrun	1205769840	8.082	4.073	5.894	**9.880**
Shuttlestart	569571680	3.825	9.965	12.007	**14.189**
Mobcal	1168846880	5.985	4.380	6.359	**9.995**
Parkrun	1255203920	5.724	4.863	7.095	**11.015**
Shields	1092507520	9.025	4.437	6.591	**12.388**
Bigships	886159200	4.871	8.331	10.922	**12.388**
City	936586320	4.952	10.123	14.626	**19.091**
Crew	725805600	4.910	7.207	10.744	**13.912**
Average	**1010851573**	**5.911**	**6.557**	**9.390**	**13.135**

Tables 5.5-5.7 show the experimental results for some lossless intra prediction methods in terms of the saving percentages of coding bits compared with the method

used in the original H.264/AVC standard (H.264_LS). Since it is a lossless coding process, there is no loss in video quality. From these tables, we can see that the method based on samplewise DPCM (H.264_DPCM) [26] has shown a large saving in terms of percentage (6.732%, 6.732%, 6.557% on average for QCIF, CIF, HDTV sequences, respectively) compare with H.264_LS. By contrast, the method based on SIDPCM [27] makes a further improvement with the modification of more intra prediction modes. For example, 8.705%, 8.346%, 9.390% of reduction in coding bits on average for QCIF, CIF, HDTV sequences have been obtained. Compared with the two methods, the proposed algorithm achieves a consistent increase in saving percentage for all tested sequences. The saving percentage is larger than 10% on average. Based on the experimental results in [26] and [29], we found JPEG_LS in [42] can achieve a better performance than motion JPEG2000 in [43] for lossless compression of video sequences. As a result, the coding efficiency of JPEG_LS in [42, 81] for the tested video sequences is evaluated by us. From the results in Tables 5.5-5.7, we can see that the JPEG_LS usually shows an improvement in saving the encoding bits compared with H.264_LS, such as 9.739% (for QCIF), 8.858% (for CIF), 5.911% (for HDTV) reduction on average of the total bits used by the H.264_LS. However, when compared with our algorithm, it can be seen that our algorithm usually saves most of the bits for the tested video sequences, especially for the sequences with high resolution.

To have a similar comparison with the method in [26], Table 5.8 gives the compression ratios of some tested video sequences encoded by using different methods compared with the raw video sequences (with 8 bits per sample). From this table, we can see that our method can achieve an average compression ratio of 1.901:1 for the tested video sequences. It is the highest comparison ratio among the five methods. Furthermore, the lossless compression efficiency of the encoder based on H.264/AVC framework can also be greatly improved when it is used together with inter-frame prediction. According to our experimental results, about 80% on average of encoding time compared with H.264_LS is increased due to the use of a 2D non-separable AIF and the rate optimization of two coding modes for residual blocks. However, it is less important for a high compression algorithm when it is heading for High Efficiency Video Coding (HEVC/H.265) [74, 75], which is the video coding standard in the near future. In addition, the separable adaptive interpolation filter [82] or the directional interpolation filter in [83] has been proved to be able to reduce the interpolation complexity compared with the non-separable adaptive interpolation filter. Therefore, it is possible that the non-separable AIF in our algorithm be substituted by some interpolation filter to reduce the interpolation complexity. However, to find the best substitution with the less complexity and the similar coding efficiency, some further investigation is required, which could be our future work. Moreover, the fast rate estimation methods, such as [84, 85] can also be combined with the proposed algorithms to speed up the optimization process of two residual coding modes, which is a fruitful direction for further research. Moreover, some other fast techniques such as those in [25, 52, 60, 61] can also be integrated. As a result, it gives some more choices to reduce the encoding complexity.

Table 5.8. Results of compression ratios

Sequence		Compression Ratio				
		H.264_LS	H.264 _DPCM	SIDPCM	JPEG_LS	**HIPAIF**
QCIF	Coastguard	1.558	1.710	1.732	1.544	**1.775**
	Monitor	1.697	1.851	1.897	1.963	**2.008**
	Foreman	1.639	1.752	1.785	1.707	**1.852**
CIF	Container	1.791	1.903	1.944	1.896	**2.009**
	Hall	1.821	1.948	1.994	2.072	**2.116**
	Coastguard	1.564	1.723	1.759	1.608	**1.803**
HDTV (720p)	Crowdrun	1.174	1.224	1.248	1.277	**1.303**
	Shuttlestart	2.485	2.760	2.824	2.584	**2.896**
	Mobcal	1.211	1.267	1.293	1.288	**1.346**
Average		**1.660**	**1.793**	**1.831**	**1.771**	**1.901**

5.4 Chapter summary

In this chapter, two algorithms for improving the lossless intra coding are proposed. In the first algorithm, a hierarchical intra prediction combined with residual coding mode optimization for lossless coding is presented. Based on the hierarchical prediction, some pixels in an MB/block are firstly predicted and encoded with the minimal rate cost, and the others are then predicted adaptively based on the minimal gradient approach using the nearer references. The proposed hierarchical prediction method makes the intra prediction more accurate, and thus the information left in the residual block can be reduced significantly. Two coding schemes are then designed to encode the residual block. The optimal one can be selected based on the rate optimization method. Furthermore, a more efficient ICAVLC entropy coding method is proposed to encode the flags of the coding schemes for an MB. Experimental results show that the proposed algorithm can improve the compression efficiency of the lossless coding in the H.264/AVC standard.

Note that the prediction method of samples in G1 to G3 can be considered as an interpolation process. Equations (5.2)-(5.4) are simple representatives of an interpolation method, and only two directions are involved in it. If a better interpolation method and multiple directions are used to do prediction, the information left in the residual block should be further reduced. Consequently, only a small number of bits may be required to encode the residual block. The compression ratio can then be further increased. Based on this consideration, the hierarchical intra prediction algorithm combined with an adaptive interpolation filter is proposed to improve lossless intra compression as the second algorithm. Since the block-based adaptive interpolation filter can learn the local structures adaptively, a better visual quality of predicted frames is thus obtained. The experimental results show that the proposed algorithm can provide a comparable predicted frame compared with other algorithms in the literature, and thereby improve the compression efficiency of the lossless coding in the H.264/AVC standard.

Note that G0 is predicted based on intra spatial prediction in H.264/AVC. In fact, we can also use other intra prediction methods to predict samples in G0. As a result, the proposed algorithm can be extended to the lossless compression in other image or video coding standards, such as high efficiency video coding (HEVC) [31] or 3D video coding [86].

It is possible that further improvement can be still obtained. This is because G0, G1, G2 and G3 are sampled interleaving from an MB. Therefore, high correlations should be appeared among these four subblocks. If the correlations among them are fully exploited, the redundancy between them will be removed largely. Some new intra prediction and coding schemes, such as those in [87-90] may be considered to improve the prediction of samples in G1 to G3.

Furthermore, the transform process can also be improved by combining the improved integer DCT in [91] or the integer sine transform proposed in [92]. As a result, intra coded frame becomes more efficient for compression. It is worth to mention that the proposed idea may have the potential to be applied to intra coded MBs for the lossy compression as a trial.

Chapter 6 Novel Adaptive Algorithm for Intra Prediction with Compromised Modes Skipping and Signaling Processes in HEVC

6.1 Introduction

The next generation of video coding standard, the High Efficiency Video Coding (HEVC) standard, has achieved significantly higher compression efficiency [93] than all the existing video coding standards due to the adoption of many advanced coding tools [31-34, 94-102]. For example, up to 35 modes are available for intra prediction of each prediction unit in HEVC. This can provide more accurate predictions and thereby improve the compression efficiency of intra coding. However, it also brings two major drawbacks. One is the encoding complexity of the intra coding method increases dramatically due to a larger number of modes involved in the mode decision process, and the other drawback is that more overhead bits are required to signal the optimal mode to be used for each prediction unit.

Intuitively, it is not necessary for all modes to be checked and signaled all the time. When the reference samples from the left-down, left, up and up-right sides are equal or parts of them are equal, the resultant predicted PUs are thereby equal regardless of the prediction modes. In this situation, only the modes which give different predicted PUs need to be checked and signaled. The other modes which give the same predicted PUs can be skipped. This strategy not only speeds up the intra mode decision process but also saves some bits consumed for signaling the mode, thereby the bitstream can be compressed. Based on this consideration, we present an adaptive intra modes skipping algorithm for decision and signaling processes in this chapter.

6.2 Adaptive Intra Modes Skipping Algorithm for Decision and Signaling

In this section, we first analyze the statistical properties of reference samples used for intra prediction. Based on the analysis, an adaptive skipping algorithm of intra modes is presented for an efficient mode decision and signaling processes.

6.2.1 Statistical properties of reference samples in intra prediction

Note that neighboring samples from the left-down, left, up and up-right are used for the prediction of the current PU. Figure 6.1 shows three typical cases of reference samples for the intra prediction of 4x4 PU in the HEVC as an example. For other PU sizes, such as 8x8, 16x16, 32x32, 64x64, the same situation can be followed. As shown in the first case of Figure 6.1, the neighboring reference samples have the same value M. So case 1 represents the current PU having a smooth reference strip. In case 2, the above and the left reference samples are the same, respectively. In other words, the current PU has a smooth reference row and a smooth reference column. In case 3, the reference samples are quite different. For the above three cases, different sets of intra prediction modes should be utilized. In other words, some modes may be skipped in different cases. The signaling of the intra prediction modes in each set is thereby re-specified correspondingly for an efficient compression.

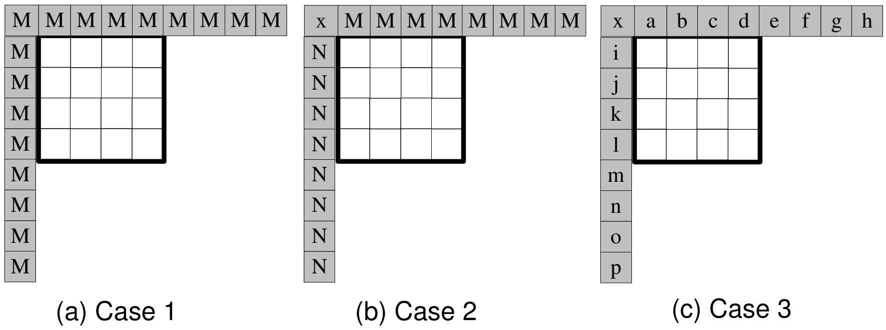


Figure 6.1. Cases of neighboring reference samples for the prediction of the current PU

Consequently, a measure of the reference smoothness becomes especially important for an efficient modes skipping algorithm. In our algorithm, the variances of reference samples defined in (6.1)-(6.3) are utilized to measure the smoothness of the whole references, the upper references and the left references, respectively.

$$Var_{all} = \frac{1}{2N+1}\left[\sum_{k=-1}^{N-1}\left(f_k^{up} - \mu_{all}\right)^2 + \sum_{k=0}^{N-1}\left(f_k^{left} - \mu_{all}\right)^2\right] \tag{6.1}$$

$$Var_{up} = \frac{1}{N}\sum_{k=0}^{N-1}\left(f_k^{up} - \mu_{up}\right)^2 \tag{6.2}$$

$$Var_{left} = \frac{1}{N}\sum_{k=0}^{N-1}\left(f_k^{left} - \mu_{left}\right)^2, \tag{6.3}$$

where μ_{all} denotes the mean value of all the reference samples in both the upper row and the left column of the current PU. The terms, μ_{up} and μ_{left}, denote the mean value of the reference samples in the upper row and the left column of the current PU, respectively. $f_k^{up/left}$ is the pixel value of the upper or left reference sample with an index k. N represents the number of reference samples in the up or the left strip. It is equal to 8, 16, 32, 64 and 128 for the PU with the size of 4x4, 8x8, 16x16, 32x32 and 64x64, respectively. To determine the smoothness, three thresholds (TH_{all}, TH_{up} and TH_{left}) are set for the three calculated variances (Var_{all}, Var_{up} and Var_{left}), respectively.

6.2.2 Adaptive intra modes skipping algorithm

Table 6.1 concludes the available intra prediction modes and the number of bits used for signaling the selected mode under different conditions for each PU.

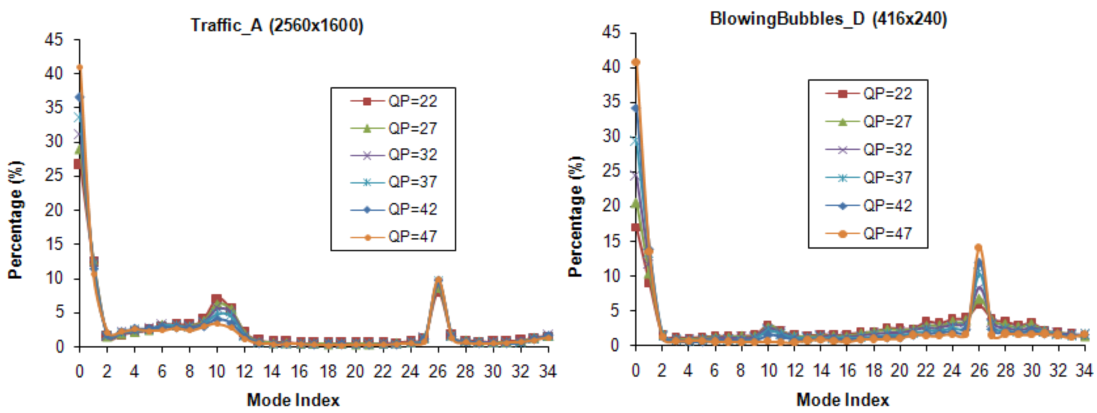
Table 6.1. Number of available intra prediction modes and bits for signaling the optimal mode under different conditions for each PU

Case	**Condition**	**Num. of modes**	**Description of modes**	**Bins for signaling the optimal mode**
1	$Var_{all} \leq TH_{all}$	1	Planar	Nil
2	$Var_{all} > TH_{all}$, $Var_{up} \leq TH_{up}$ and $Var_{left} \leq TH_{left}$	19	Planar, DC, Hor-i ($0 \leq i \leq 7$), and Ver-i ($0 \leq i \leq 8$)	2~5
3	Otherwise	35	Planar, DC, Hor+x ($-7 \leq i \leq 8$), and Ver+i ($-8 \leq i \leq 8$)	2~6

Referring to Figure 6.1 and Table 6.1, the details of the three cases are described as follows:

(1) Case 1

This case represents that all the references are the same or very similar. In this case, we use Var_{all} to measure the smoothness of all the references. If Var_{all} is smaller than TH_{all}, the references are considered smooth. The 35 predicted PUs are always the same or quite similar regardless of the prediction modes. In such a case, only one mode is enough for predicting the current PU. Figure 6.2 shows the percentage of each mode selected after mode decision for two classes of videos based on the mode decision algorithm in HM7.0. From Figure 6.2, we can see that the Planar mode always has the largest probability to be the best mode. As a result, only the Planar mode is adopted to predict the current PU when the reference samples are consistently smooth. Due to only one mode employed, there is no mode decision process, and no overhead bits are required to signal the mode. In the decoder side, Var_{all} is firstly calculated. Planar mode is used to reconstruct the current PU if Var_{all} is smaller than TH_{all}. This case not only significantly reduces the encoding complexity, but also saves the overhead bits used for signaling the best intra prediction mode.



(a) (b)

Figure 6.2. Percentages of the selected modes under six QP values

(2) Case 2

If Case 1 is not satisfied, i.e. Var_{all}>TH_{all}, Case 2 is checked. In this case, the above and the left reference samples are the same as shown in Figure 6.1, respectively. The predicted blocks produced by modes {Ver+i| i=0,1,...,8} or {Hor+i| i=0,1,...,8} are the same, respectively. Therefore, it is not necessary to use all the intra prediction modes. In order to measure the smoothness of the reference row and the reference column, both Var_{up} and Var_{left} are calculated by using (6.2) and (6.3), respectively. If Var_{up}<TH_{up} and Var_{left}<TH_{left}, only 19 modes, i.e. Planar, DC, {Hor-i| i=0, 1,...,7} and {Ver-i| i=0, 1,...,8}, are required as shown in Table 6.1 instead of 35 modes. For the 19 intra prediction modes, the RMD plus RDO method is used to select the optimal mode for the current PU. During the RMD process, the number of candidate modes are the same as those in HM7.0, i.e. 8, 8, 3, 3, 3 for 4x4, 8x8, 16x16, 32x32, 64x64, respectively. Furthermore, the three MPMs specified by HM7.0 are always included in the candidate list. Due to our arrangement that only 19 modes (0,1, 10, ..., 26) are available in case 2 for the current PU, it is necessary to map the MPMn (n=0,1,2) to one of the 19 modes when we append it in the candidate list if the MPMn (n=0,1,2) is out of those 19 modes. The mapping is shown as in Table 6.2 (a) for the mode decision process.

Table 6.2. Mapping of intra prediction modes for Case 2 in the encoding process

(a) Mapping of the MPMn (n=0,1) when appending it to the candidate list for mode decision process					
MPMn (n=0,1)	0	1	2, ..., 9	10, ..., 26	27, ..., 34
Mapped MPMn (n=0,1)	0	1	10	10, ..., 26	26
(b) Mapping of the current mode when signaling it					
CurMode	0	1	x	10, ..., 26	x
Mapped CurMode	0	1	x	2, ..., 18	x
(c) Mapping of the MPMn (n=0,1,2) when signaling the CurMode					
MPMn (n=0,1,2)	0	1	2, ..., 10	11, ..., 25	26, ..., 34
Mapped MPMn (n=0,1,2)	0	1	2	3, ..., 17	18

Similar to the strategy in HM7.0, three MPMs (MPM0, MPM1 and MPM2) are utilized for signaling the intra prediction modes for achieving a good coding efficiency in Case 2. The detailed signaling (codewords) of CurMode is shown in Table 6.3. When the CurMode is among the three MPMs, a bypass mode (assuming an equal probability distribution) or an adaptive context algorithm (assuming an unequal probability distribution) is adopted to encode prev_intra_luma_pred_flag (a flag used to signal if the CurMode is among the three MPMs) and mpm_idx (mpm_idx is signaled to indicate whether CurMode is MPM0, MPM1 or MPM2). When the CurMode is not equal to any MPM, the rem_intra_luma_pred_mode which denotes the left mode is binarized using a FLC (0xxxx) as shown in Table 6.3. Note that only 19 modes instead of 35 modes are required to signal. Therefore, when prev_intra_luma_pred_flag is 0, four bits are enough to signal the left modes (16 modes). Note that the mode indexes of the 19 modes are 0, 1, 10, ..., 26. In order to signal it efficiently into the bitstream, the 19 modes are mapped to consecutive mode numbers from 0 to 18 as shown in Table 6.2 (b). Correspondingly, the three MPMs should be mapped to have a consistent indexing to the 19 intra prediction modes as shown in Table 6.2 (c), when signaling the CurMode. We label the three mapped MPMs as MMPMn (n=0,1,2). One point that we would like to stress is that the three MPMs are totally different before mapping. However, they may be mapped to the same mode based on Table 6.2 (c). In such a case, we must convert them to three different and decodable MMPMs according to the following method.

```
If (MMPM0==MMPM1)
{
        If (MMPM0>1)
        {
                MMPM1=(MMPM0+14)%17+2;
                MMPM2=(MMPM0-1)%17+2;
        }
        Else
        {
                MMPM0=0;
                MMPM1=1;
                MMPM2=18;
        }

}
Else
{
        If (MMPM0 && MMPM1)
                MMPM2=0;
        Else
                MMPM2=(MMPM0+MMPM1)<2?18:1;
}
```

In the decoder side, the same operation should be performed to decode the current mode when Case 2 is detected for the current PU.

Table 6.3. Codewords for intra Luma prediction modes in Case 2

prev_intra_luma_pred_flag	**Codewords**
1	If CurMode=MPM0, Codewords=10; If CurMode=MPM1, Codewords=110; If CurMode=MPM2, Codewords=111.
0	Codewords=0xxxx (Note: xxxx is used for binarizing **rem_intra_luma_pred_mode**, which is from 0000 to 1111)

(3) Case 3

If both the first and the second conditions are not satisfied, the reference samples are considered as having large variations. In this situation, all the 35 intra prediction modes are involved in the normal mode decision process. The same operation as that in HM7.0 is performed for this case.

6.2.3 Determination of thresholds for intra modes skipping

With further experimental work, we have found that TH_{all} should be made adaptive to the quantization parameter (QP), texture of encoded video sequence and bit depth of Luma components. As a result, TH_{all} can be obtained by (6.4).

$$TH_{all} = w_{QP} \times w_{Texture} \times w_{Bit\,Depth}, \tag{6.4}$$

where w_{QP} specifies the influence of the quantization step size Qstep which is indexed by QP. In our experiments, we used BlowingBubbles in Class D with the smallest resolution as a tested sequence. By trying a large number of thresholds, we found that the rate distortion performance of the proposed algorithm for BlowingBubbles is better than that in HM when w_{QP} is defined in (6.5) for All Intra

Main configuration. In fact, it can be set to a larger value to form a fast algorithm in real-time applications when a slight loss of rate distortion performance is permitted.

$$w_{QP} = \left\lfloor \frac{Qstep^2}{61} + 0.5 \right\rfloor \tag{6.5}$$

In order to make it suitable for other input sequences, $w_{Texture}$ defined in (6.6) is also used to adjust the threshold. The value of $w_{Texture}$ can be considered as a constant for a short time period, especially when there is no abrupt shot transition. In (6.6), p_s denotes the probability that the Planar mode is selected in 64x64 PUs during the RDO process for sequence s, and p_0 denotes the probability of the Planar mode selected when BlowingBubbles is the tested sequence. The value of p_s can be calculated by two methods. One possible method to calculate p_s is to use the original method in HM for encoding the first frame of the tested sequence. The number (N_{Planar}), for which the Planar mode is selected as the best intra prediction mode, is used to form the statistics. There is no further calculation required for the rest of the sequence. N_{Planar} is then used to calculate p_s as shown in (6.7) in the following frames.

$$w_{Texture} = \frac{p_s}{p_0} \tag{6.6}$$

$$p_s = \frac{N_{Planar}}{N_{Total}} \times 100\% \tag{6.7}$$

In (6.7), N_{total} is the number of PUs in the tested frames. To have a better coding performance, $w_{Texture}$ is updated after a number of frames have been encoded. The number of coded frames is decided by the users. For the first method, p_0 is 34 based on our simulation. Another feasible method used to calculate p_s is that we can initially determine it offline by encoding some test video sequences using a larger QP value, such as QP=51. If the first method is used to calculate p_s, there is no extra syntax element required in the bitstream. If the second method is used to obtain p_s, parameter p_s should be encoded into the bitstream by using only seven bits for each video sequence to inform the decoder.

To make it be adaptive to the All Intra HE10 configuration, parameter $w_{Bit\ Depth}$ is used to adjust the threshold to an image with different bit depths (n). It is calculated as follows:

$$w_{Bit\ Depth} = \frac{\left(2^n - 1\right)^2}{\left(2^8 - 1\right)^2} \tag{6.8}$$

Compared with the method in HM7.0, we can see that our method can save about 2 to 6 bits if Case 1 is selected as shown in Tables 2.2 and 6.3. By contrast, only 1 bit may be saved by using our method if Case 2 is selected. This means that Case 2 plays a less important role than Case 1 in terms of overhead bits saving. Furthermore, both the left strip and the upper strip are smooth if Condition 1 happens. In other words, part of Case 2 has been included in Case 1. For the remaining part, i.e., $|M - N| > 2\sqrt{TH_{all}}$, we employ a condensed region to detect it as Case 2. Note also that the number of references in the left or upper side of the current PU is half number of all the references. Based on this consideration and our experimental work, the condition in Case 2 is set as $TH_{up}=TH_{all}/2$ and $TH_{left}=TH_{all}/2$ at present.

6.2.4 Flowchart of adaptive intra modes skipping algorithm in HM

Figure 6.3 summarizes the improved intra coding process of a PU. Given an intra PU with the size of 64x64, 32x32, 16x16, 8x8 or 4x4, the variance (Var_{all}) of all the neighboring samples including the left-down, left, up, up-right regions is calculated based on (6.1). If it is smaller than TH_{all}, only Planar mode is used to predict the current PU. Both the mode decision (MD: RMD plus RDO) and mode signaling (MS) processes are skipped. The residual PU between the current PU and the predicted PU is performed by the RQT technique [36, 37] to select the optimal

TUs. Finally, entropy coding is performed to encode all the intra related information for the current PU. If not, the variances (Var_{left} and Var_{up}) of reference samples in the left and upper strips are calculated, respectively. If both $Var_{up} \leq TH_{up}$ and $Var_{left} \leq TH_{left}$ are satisfied, 19 intra prediction modes are set for the RMD and RDO processes. Otherwise, all the 35 intra prediction modes are checked in the normal intra coding process in HM7.0.

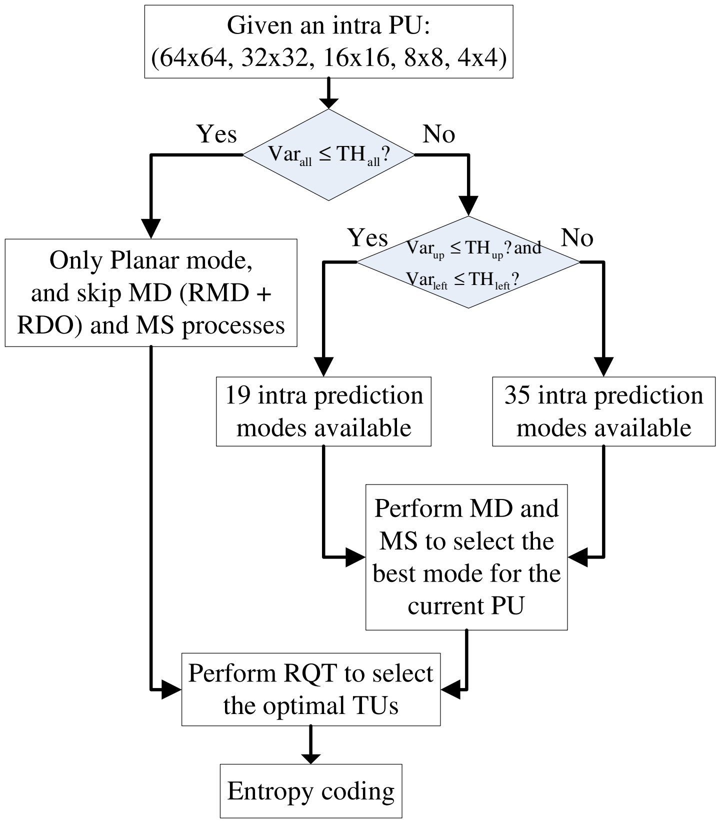


Figure 6.3. Flowchart of adaptive intra modes skipping algorithm based on HM7.0

6.2.5 Decoding of intra prediction mode

In the decoder side, Var_{all} is firstly calculated for the current PU. If Var_{all} is smaller than TH_{all}, the decoding process of intra prediction mode is skipped, and the Planar mode is used to predict the current PU. Otherwise, Var_{left} and Var_{up} are calculated to check whether Case 2 is satisfied. If Case 2 is satisfied, three MPMs are obtained based on the prediction modes of neighboring decoded PUs. In accordance with the encoder side, each MPMn (n=0,1,2) should be mapped to one of the 19 indexes according to Table 6.4 (a). The prev_intra_luma_pred_flag is decoded firstly to determine whether the current mode is one of the three MPMs. If it is, decoding of the current mode is the same as that in HM7.0. Otherwise, only four bits are decoded to parse the current prediction mode. Note that the decoded mode number ranging from 0 to18 is not the real mode index used for the prediction of the current PU. It should be mapped to its original mode index defined in HM7.0 according to Table 6.4 (b) in order to predict the current PU. If both Case 1 and Case 2 are not satisfied, the current PU belongs to Case 3. The decoding process of it is the same as that in HM7.0.

Table 6.4 Mapping of intra prediction modes for Case 2 in the decoding process

(a) Mapping of the MPMn (n=0,1,2) when decoding the CurMode					
MPMn (n=0,1,2)	0	1	2, ..., 10	11, ..., 25	26, ..., 34
Mapped MPMn (n=0,1,2)	0	1	2	3, ..., 17	18
(b) Mapping of the decoded mode when storing it for prediction					
Decoded CurMode	0	1	2, ..., 18	x	
Mapped CurMode	0	1	10,...,26	x	

Figure 6.4 shows an example of partitioning a 64x64 LCU into various sizes of PUs which are determined by an HEVC encoder. The numbers from 0 to 12 indicate the processing orders including encoding order and decoding order. To reconstruct a LCU in a decoder side based on HM7.0, the intra prediction modes (Mn, for n from 0 to 12) and the DCT coefficients (Cn, for n from 0 to 12) of residual PUs need to be decoded. The decoding order of Mn and Cn is M0->C0->M1->C1->...->M12->C12

in this example. After all the modes and DCT coefficients have been decoded, the decompression process is called to reconstruct the LCU based on the order, B0->B1->...->B12 according to HM7.0. Note that in our algorithm, to determine the number (1, 19 or 35) of available intra prediction modes for each PU, the neighboring PUs should be reconstructed when we parse the intra prediction mode of the current PU. Therefore, the decoding order is adjusted as M0->C0->B0->M1->C1->B1->...->M12->C12->B12.

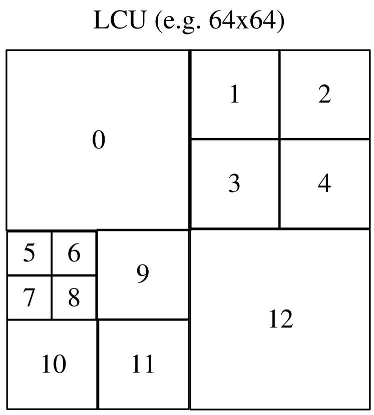


Figure 6.4. Example about the processing order of PUs in a LCU

6.3 Experimental Results

The proposed algorithm has been implemented in HM7.0 software [35] of the HEVC test model. Since we gave focus on the improvements of intra coding efficiency, all the frames were intra coded. Two encoding configurations for intra coding: AIMain and AIHE10 described in [44] were used. The proposed algorithm was compared against HM 7.0 without the proposed algorithm (the anchor). Six QP values: 22, 27, 32, 37, 42 and 47 were tested, which are in the mid-bitrate and low-bitrate ranges. Five video sequences (Traffic, ParkScene, RaceHorsesC, BlowingBubbles and Vidyo4) from five classes with different resolutions as specified in HEVC [44] are encoded. All frames were used for each tested video sequence. The rate distortion performances of the proposed algorithm for mid-bitrate and low-bitrate ranges were calculated based on the method in [50]. The reduction of encoding time (ΔTIME) with respect to HM7.0 was measured. In the following tables, positive values mean increase, and negative values mean decrease. The experimental work was done on a platform with CPU of Intel Xeon 5460 @3.16 GHz and RAM of 32.0GB.

Table 6.5. Simulation results in terms of BD-Rate in mid-bitrate range

Seq.	All Intra Main (%)		All Intra HE10 (%)	
	Comp1	Comp2	Comp1	Comp2
Class A	-0.10	-0.15	-0.11	-0.17
Class B	-0.09	-0.16	-0.10	-0.18
Class C	-0.03	-0.05	-0.03	-0.05
Class D	0.01	-0.01	0.01	-0.01
Class E	0.02	-0.10	-0.10	-0.21
Average	-0.04	-0.09	-0.07	-0.12

Table 6.6. Simulation results in terms of BD-Rate in low-bitrate range

Seq.	All Intra Main (%)		All Intra HE10 (%)	
	Comp1	Comp2	Comp1	Comp2
Class A	-0.20	-0.42	-0.18	-0.45
Class B	-0.17	-0.67	-0.11	-0.57
Class C	-0.21	-0.43	-0.16	-0.35
Class D	-0.25	-0.42	-0.19	-0.33
Class E	-0.09	-0.61	-0.25	-0.72
Average	-0.18	-0.51	-0.18	-0.48

The BD-Rate performances are shown in Tables 6.5 and 6.6 for mid-bitrate (with the values of QP: 22, 27, 32 and 37) and low-bitrate ranges (with the values of QP: 32, 37, 42 and 47), respectively. For each configuration, two comparisons

(Comp1 and Comp2) are made. "Comp1" means the comparison between our algorithm and the HM 7.0. "Comp2" means the comparison between our algorithm and the modified HM 7.0. Note that the only difference between the modified HM7.0 and the HM7.0 is that the prev_intra_luma_pred_flag is coded by making use of the bypass mode in the modified HM7.0 instead of an adaptive context algorithm used in HM7.0. Correspondingly, the prev_intra_luma_pred_flag was coded by using an adaptive context algorithm in our algorithm for Comp1, and it was coded with the bypass mode in our algorithm for Comp2.

From the results, we can see that the proposed algorithm shows a slight decrease in BD-Rate in the mid-bitrate range compared with the HM7.0 (Comp1). However, the BD-Rate is reduced consistently compared with the modified HM7.0 for the mid-bitrate range. An average of 0.11% BD-Rate decrease (Comp2) is achieved for both All Intra Main and All Intra HE10 in the mid-bitrate range. By contrast, the proposed algorithm has better compression efficiency in low-bitrate range for both Comp1 and Comp2. For All Intra Main configuration, 0.18% and 0.51% BD-Rate improvements are achieved on average compared with the original HM7.0 and the modified HM7.0, respectively. For All Intra HE10 configuration, 0.18% and 0.48% BD-Rate improvements are achieved on average for Comp1 and Comp2, respectively. All experimental results in Tables 6.5 and 6.6 illustrate clearly that the proposed algorithm shows better compression efficiency as compared to that in HM7.0 of HEVC for the low-bitrate range.

Besides the benefits in terms of rate distortion performance, the proposed algorithm reduces significantly the encoding time due to the skipping of some unnecessary intra prediction modes compared with that in HM7.0. As shown in Table 6.7, the timesaving is increased with an increase of QP values. In the low-bitrate range, on average 7.75%, 14.40%, 26.03% and 37.93% of timesaving were achieved corresponding to QP values of 32, 37, 42 and 47 in All Intra Main configuration compared with the HM7.0. For the All Intra HE10 configuration, on average 7.57%, 13.81%, 23.78% and 34.11% in time reduction were achieved for the low-bitrate ranges. However, the timesaving is not obvious when QP is smaller. Sometimes, the encoding time even increases slightly compared with that in HM7.0. This is because a smaller value was used as the threshold (TH_{all}) in (6.4) for smaller QP values (high-bitrate range). In contrast, the encoding complexity can be reduced significantly if larger thresholds are given to TH_{all}. In such a case, the proposed algorithm is changed to a purely fast intra mode decision algorithm without or with less benefits to the rate distortion performance. For example, we can reduce the 35 modes to 1 or 19 by comparing the variance with a larger threshold before the intra mode decision process in the encoder side of HM for high-bitrate range, while the signaling processing of the best intra mode is the same as that of HM. In another word, we still signal the best intra mode according to the case with 35 available modes. As a result, the encoding complexity can be reduced significantly, and the decoding process is the same as that in HM.

Table 6.7. Simulation results in terms of times consuming for the encoder

Conf.	All Intra Main [%]						All Intra HE10 [%]					
QP	22	27	32	37	42	47	22	27	32	37	42	47
Class A	2.16	-0.83	-7.83	-15.05	-25.74	-36.47	0.97	-1.97	-7.19	-14.26	-23.02	-32.3
Class B	2.06	-2.58	-8.71	-17.14	-30.15	-44.33	0.12	-2.96	-8.32	-15.81	-27.41	-39.26
Class C	1.44	-1.07	-5.54	-13.05	-29.14	-39.55	0.7	-1.02	-5.59	-11.92	-26.39	-35.5
Class D	1.61	1.49	0.85	-1.94	-11.46	-29.03	1.63	1.48	0.84	-2.99	-10.88	-25.82
Class E	-0.49	-7.89	-17.53	-24.83	-33.65	-40.25	-4.98	-10.62	-17.57	-24.07	-31.19	-37.67
Average	1.36	-2.18	-7.75	-14.40	-26.03	-37.93	-0.31	-3.02	-7.57	-13.81	-23.78	-34.11

Table 6.8. Simulation results in terms of times consuming for the decoder

Conf.	All Intra Main ($Time_{HM7.0}/Time_{Proposed}$)						All Intra HE10 ($Time_{HM7.0}/Time_{Proposed}$)					
QP	22	27	32	37	42	47	22	27	32	37	42	47
Class A	1.07	1.08	1.07	1.07	1.07	1.05	1.06	1.05	1.05	1.06	1.06	1.09
Class B	1.07	1.07	1.07	1.07	1.07	1.06	1.06	1.06	1.05	1.07	1.02	1.08
Class C	1.07	1.07	1.07	1.07	1.08	1.07	1.05	1.06	1.06	1.05	1.05	1.07
Class D	1.09	1.09	1.08	1.09	1.11	1.09	1.07	1.07	1.07	1.06	0.87	0.99
Class E	1.07	1.07	1.06	1.06	1.07	1.05	1.05	1.02	1.04	1.03	1.06	1.04
Average	1.07	1.07	1.07	1.07	1.08	1.06	1.06	1.05	1.05	1.05	1.01	1.05

In order to decode the intra prediction mode, the variance of the neighboring reference samples for each PU should be calculated in the decoder side. This increases the decoding time which is shown in Table 6.8 for the two configurations. From the results, we can see that the proposed algorithm has little influence on the complexity of the decoder. An average of 6% increase in decoding time based on HM7.0 [35] is required. Due to the unsymmetrical design of the encoder and decoder for HEVC, the decoder time is only 1.5% to 1.6% of the encoder. Hence a small percentage increase in decoding time gives litter effect to the decoder. Furthermore, we would like to leave this point as an open question, and the realization of this paper is just a sample realization. The ideas in the paper can be improved further by investigating simpler and more accurate thresholds making use of variance again or other simple features. On the other hand, if we do not want to increase the computational burden of the decoder side, we can encode the case index among the three cases (Case 1, Case 2 and Case 3 as discussed in Section III) into the bitstream for each PU in the encoder side. The decoder can then decode it. In this case, our algorithm becomes a fast algorithm. No additional computational complexity is required for the decoder. The ideas in this paper can provide a new thread of thought. According to various applications and requirements, better improvements or realization techniques can be further investigated, which is a fruitful direction for interested researchers.

6.4 Chapter summary

In this chapter, a novel adaptive intra modes skipping algorithm for mode decision and signaling processes in HEVC is presented. By making full use of the statistical properties of the neighboring reference samples, the proposed algorithm can adaptively optimize the set of intra prediction modes involved in mode decision and signaling processes. As a result, the proposed algorithm not only saves the bits for signaling the overhead, but also accelerates the intra mode decision process. Experimental results show that the proposed algorithm can provide a better rate distortion performance than that in HM7.0 for low-bitrate ranges. In a meanwhile, the proposed algorithm can significantly reduce the complexity of intra coding for both All Intra Main and All Intra HE10 configurations for low-bitrate ranges.

Moreover, the proposed algorithm has a wide range of adaptability. It can be integrated with other algorithms, such as [38-41, 103] which are directly related to the intra coding in HM, or [22-25, 60, 61, 91] which have the potential to be extended to HM, to further improve the rate distortion performance and reduce the encoding complexity of these algorithms.

It is possible that further improvement can still be obtained. Note that the threshold in (6.4) is only adaptive to the whole video sequence. In fact, different thresholds are preferred for different sizes of PUs (64x64, 32x32, 16x16, 8x8 and 4x4) in order to achieve a better rate distortion performance. As a result, it is also possible for high-bitrate range to achieve better compression efficiency than that in HM. Furthermore, to achieve better compression efficiency with reduced encoding complexity than HM, other features can also be used as detectors instead of the variance in (6.1)-(6.3) to determine the possible number of intra prediction modes for each PU. This is a fruitful direction of future research work.

Chapter 7 Conclusions

7.1 Conclusion on the current works

The current H.264/AVC is a matured international video coding standard. It has a very broad application range that covers all forms of digital compressed video from low bit-rate internet streaming applications to HDTV broadcast and Digital Cinema applications with nearly lossless coding. It can provide the higher compression efficiency than previous standards (i.e. about half of bit rate savings than MPEG-2, H.263, or MPEG-4 Part 2) [1] with good video quality due to the adoption of many advanced video coding tools. As a result, the encoding complexity of an encoder based on the H.264/AVC standard is higher for real time applications. In this thesis, we have proposed several techniques to reduce encoding complexity of an encoder based on H.264/AVC. On the other hand, with the higher resolution videos gradually entering into our daily life, to further improve the compression efficiency of H.264/AVC is still required. Therefore, we have also developed a few techniques to enhance the coding efficiency based on the H.264/AVC and the coming HEVC standards. Conclusions about these proposed techniques are described as follows.

7.1.1 Techniques to reduce the encoding complexity

In Chapter 3, we propose two early detection algorithms for ZQDCT coefficients of inter residual blocks. Compared with the conventional detection methods of ZQDCT coefficients used in H.264/AVC, the first algorithm has two major features. First, a new classification of patterns for DCT, Q, IQ and IDCT processes is suggested based on the zigzag scanning order. By using the new classification, the quantized DCT coefficients can be coded efficiently. Second, the required thresholds for detecting ZQDCT coefficients are determined by combining the Gaussian distribution with a theoretical analysis of DCT and quantization in H.264/AVC. Experimental results show that the proposed algorithm can achieve an average timesaving of more than 40% for DCT, Q, IQ and IDCT processes, compared with the algorithm used in JM12.2. When compared with other algorithms in the literature, it also gives the best performance in terms of both rate-distortion and time saving. By making use of the new classification of patterns for DCT, Q, IQ and IDCT processing, the second algorithm which can give different thresholds for the detection of ZQDCT coefficients in H.264/AVC is then proposed. In the second algorithm, a joint probability model is firstly developed based on the distribution of residual samples and the conditional distribution of DCT coefficients. A Lagrangian FAR-FRR optimization method is then proposed to tradeoff between the quality of the reconstructed videos and the encoding complexity. The threshold for each ZQDCT coefficient is obtained during the process of FFO. Given different values to the Lagrangian multiplier, different thresholds can be obtained. Different reconstructed video quality and different encoding complexities are thereby provided by using different thresholds. For example, if we want to save more computational complexity during the encoding process for real-time application, a larger value of Lagrangian multiplier should be set. The saving of the computation in software coders corresponds to a reduction of the power consumption in hardware coders, which is also in line with the concept of green computation advocated recently. On the contrary, if better reconstructed video quality is required, a smaller Lagrangian multiplier is preferred.

In Chapter 4, we propose some techniques to speed up the process of intra prediction based on the H.264/AVC standard. First, the candidate intra prediction types for an intra coded MB can be determined by an improved variance-based strategy. In this strategy, an adaptive threshold is adopted based on the quantization parameter. This makes the threshold more reasonable, and thereby the determination of intra prediction type is more accurate. Second, a directional difference measure is then applied to select the better candidate prediction modes for each candidate prediction type. This strategy critically touches upon the intrinsic property of the way H.264/AVC makes the intra mode prediction. Furthermore, an early termination strategy is suggested for RDO calculation. Experimental results show that the proposed algorithm by combining all the three techniques can achieve more than 80% timesaving compared with that used in JM12.2 at the expenses of a slight decrease in PSNR and a slight increase in bit rate. Compared with other algorithms in the literature, the proposed algorithm obviously shows the best performance among all the three measures.

7.1.2 Techniques to improve the rate distortion performance

Note that the lossless coding technique is also integrated into the H.264/AVC standard. With the demand for higher fidelity of encoded videos, it is possible that either the whole picture or some regions of a picture in a video need to be represented without any loss of fidelity. However, the lossless coding technique used in H.264/AVC is not efficient. Note that the intra block/MB is predicted as a whole by an extrapolation of the neighboring reconstructed pixels in the H.264/AVC standard. As a result, the samples which are far from their references may not be predicted well due to poor correlation. Moreover, the entropy coding in H.264/AVC is generally designed to encode the quantized DCT coefficients of the residual block.

It is usually not efficient to encode the residual block directly. In order to resolve these problems, we propose two algorithms in Chapter 5 to improve the lossless intra coding in H.264/AVC. In the first algorithm, a hierarchical intra prediction algorithm combined with residual mode optimization is proposed. In this algorithm, we suggest that samples in a block or MB should be predicted hierarchically instead of using the whole-block based method. More specifically, four groups are extracted from the samples in a block/MB. Based on the hierarchical prediction, samples in the first group are firstly predicted and encoded with the minimal rate cost, and the samples in the left three groups are then predicted adaptively based on the minimal gradient approach using the nearer references. The proposed hierarchical prediction method makes the intra prediction more accurate due to the adoption of nearer references as the reference, and thus the information left in the residual block can be reduced significantly. Two coding schemes are then designed to encode the residual block. The optimal one can be selected based on the rate optimization method. Furthermore, a more efficient ICAVLC entropy coding method is proposed to encode the flags of the coding modes for an MB. Experimental results show that the proposed algorithm can improve the compression efficiency of lossless intra coding algorithm used in JM12.2. Note that the prediction method of samples in G1 to G3 can be considered as an interpolation process. In the first algorithm, a simple linear interpolation based on intensity gradient measure is used to predict the samples in G1, G2 and G3. Only two directions are involved in it. If a better interpolation method with multiple directions is used to do prediction, the information left in the residual block should be further reduced. Consequently, only a small number of bits may be required to encode the residual block. Based on this consideration, the hierarchical intra prediction algorithm combined with an adaptive interpolation filter is proposed to improve lossless intra compression as the second algorithm. In this algorithm, the block-based adaptive interpolation filter is adopted to predict the samples in G1, G2 and G3. Since the block-based AIF can learn the local structures adaptively, a better visual quality of predicted frames is thus obtained. The experimental results show that the proposed algorithm can provide a comparable predicted frame compared with other algorithms in the literature, and thereby improve the compression efficiency of the lossless coding in the H.264/AVC standard. On average, about 13.485%, 11.803% and 13.135% of bitrate saving are achieved for the tested QCIF, CIF and HDTV sequences compared with JM18.0, respectively. When compared with other algorithms in the literature, the proposed algorithm also shows the best compression efficiency.

With the high resolution videos gradually entering into our daily life, the demand for further improving the compression efficiency of H.264/AVC continues to be required. The next generation standard of video coding, HEVC, is thus being developed aiming at substantially improving coding efficiency of H.264/AVC High Profiles, i.e. to reduce bitrate by half with comparable image quality, at the expense of increased computational complexity. In the current HEVC, up to 35 modes are available for intra coding. As a result, not only the encoding complexity is increased, but also more overhead bits are required to signal the mode. However, we have found that there is still room for improvement in terms of intra compression efficiency with less encoding complexity. Intuitively, not all the 35 intra prediction modes are always required to be checked and signaled. Therefore, we propose an adaptive modes skipping algorithm for mode decision and signaling processes in Chapter 6. Note that the neighboring reconstructed samples are used as the reference to predict the current PU. Based on the statistical properties of the reference samples, the proposed algorithm is able to adaptively optimize the set (with 1 mode, 19 modes or 35 modes) of intra prediction modes in each PU for the mode decision and signaling processes. As a result, not only the overall coding efficiency can be improved, but also the encoding time is reduced significantly due to less number of modes involved. More important, the proposed algorithm has the potential to be integrated with some algorithms related to intra coding in HEVC to further improve them in terms of both rate distortion performance and encoding complexity. Experimental results show that, compared to the test model HM7.0 of HEVC, BD-Rate savings of 0.18% and as well as 0.18% on average are achieved for AI-Main and AI-HE10 cases for low-bitrate ranges, and the average encoding times can also be reduced by 8%-38% and 8%-34% for AI-Main and AI-HE10 cases in low-bitrate ranges, respectively.

7.2 Directions of future research

It has been about ten years since H.264/AVC was ratified as an international video coding standard in 2003. In the past ten years, a great many research works have been done to improve an encoder based on the H.264/AVC standard in terms of coding efficiency or coding complexity. As a result, it can provide good video quality with substantially lower bitrates than previous standards (i.e. half or less the bit rate of MPEG-2, H.263, or MPEG-4 Part 2). To reduce the encoding complexity of an encoder based on H.264/AVC was an active area of research in the past few years, such as fast intra [22-25, 60-61, 64-65] or inter mode decision [104-107], fast motion estimation [108-112], early detection of ZQDCT coefficients [12-21], fast rate distortion estimation [113-118] and so on. All these techniques make an encoder based on the H.264/AVC standard possible for real time applications. Hence the H.264/AVC standard has been widely adopted by the industry. Despite its success, improving the compression efficiency of H.264/AVC continues to be required, especially for the videos with ultrahigh definition (UHD) resolution. This is because the current internet and broadcasting networks do not even have sufficient capacity to transmit large amounts of HD content, let alone UHD [119]. In this situation, HEVC, the next generation of video coding standard, has been developed by JCT-VC formed in 2010, with the aim to significantly improve the compression efficiency of the H.264/AVC high profile. The JCT-VC group has integrated a number of advanced technologies into the test model HM [35] of the HEVC standard since the first meeting in April 2010. Currently, the HEVC (based on HM5.1 software) has shown the better efficiency than H.264/AVC standard (based on JM16.2 software) in terms of coding efficiency (about 29.14-45.54% bits reduction or 1.4-1.87dB PSNR increase compared with H.264/AVC [119]). The HEVC will be ratified as an international video coding standard in 2013. Before the HEVC standard is finished, we will put a lot of effort on the improvement of compression efficiency for HEVC as the recent direction of our research. Note that the encoding complexity of an encoder based on HEVC is extremely high due to the integration of some new techniques, such as flexible block structure from 64x64 to 8x8 with recursive quadtree partitioning, more intra prediction modes (one Planar mode, one DC mode and 33 directional modes), residual quadtree based transforms ranging from 32x32 to 4x4 sizes and so on. When the HEVC standard has sufficient capability to compress video material with high or ultrahigh definition resolution, developing fast algorithms to reduce the coding complexity of the encoder with negligible loss in coding efficiency is becoming increasingly important for real time applications. This will be another important direction of our research in future.

7.2.1 To improve intra coding efficiency of an encoder based on HEVC

Note that the two hierarchical intra coding algorithms based on intensity gradient measurement and two-dimensional adaptive interpolation techniques are presented in Chapter 5 for lossless compression. In fact, the two algorithms can also be applied to lossy video coding based on the transformation and quantization techniques to improve the block based intra prediction in H.264/AVC. Moreover, all of these techniques can be extended to the new coming HEVC standard to further improve the intra coding efficiency. Especially for the large prediction units, such as 64x64, 32x32, 16x16, 8x8, the inaccurate prediction should be more obvious for the samples away from their references. In such a case, the HIP algorithm demonstrates its superiority. However, many problems still remain to be investigated before the HIP algorithm could be extended into HEVC. In intra coding of the HEVC standard, a three-stage approach which has been included in Section 2.4.1 was implemented in the test model HM to find the best intra prediction mode and the optimal transform structure. To extend the HIP algorithm to intra prediction in HEVC, we took PU with size of 32x32 as an example as shown in Figure 7.1 (a). By using the HIP algorithm, the 32x32 PU is sampled into four subblocks from G0 to G3 as shown in Figure 7.1 (b). Samples in G0 are firstly predicted, encoded and reconstructed by using the above three-stage intra coding approach. The reconstructed G0 is then used as the low resolution block for predicting samples in G1, G2 and G3 by utilizing the intensity gradient measure or adaptive interpolation techniques. Consequently, the information left in the residual G1, G2 or G3 could be reduced due to nearer samples used as the references and less bits are required to encode the residual information. Different from the prediction of G0, there were no rough mode decision and no RDO process for G1, G2 and G3. However, the RQT process for G1, G2 and G3 is still required for an efficient transform coding. Note that three TU sizes as shown in Figure 7.1 (c) for each PU in intra coding are possible. If the depth of TU is also set to three as shown in Figure 7.1 (d) for each extracted subblock (G0, G1, G2 or G3), four RQT structures corresponding to the four extracted subblocks should be signaled using a recursive depth-first approach. By contrast, only one RQT structure is required to signal for the 32x32 PU. Moreover, the coded block flag (CBF) to indicate whether there are non-zero transform coefficients in each TU is signaled. As a result, the overhead bits will cause some impacts on the total bitrates although the residual information is still reduced due to the use of more accurate prediction algorithms. In order to reduce the overhead bits used for RQT structure and CBF, one possible way is to reduce the depth of TU for each extracted subblock to two as shown in Figure 7.1 (e). In such a case, the overhead bits used for signaling the RQT structure and CBP can be reduced significantly compared with the RQT depth of three. The encoding complexity should also be decreased. However, less flexibility is then provided for choosing the transform size. The setting of TU depth is a crucial issue that should be investigated before the HIP can be extended successfully into the intra coding in the coming video coding standard.

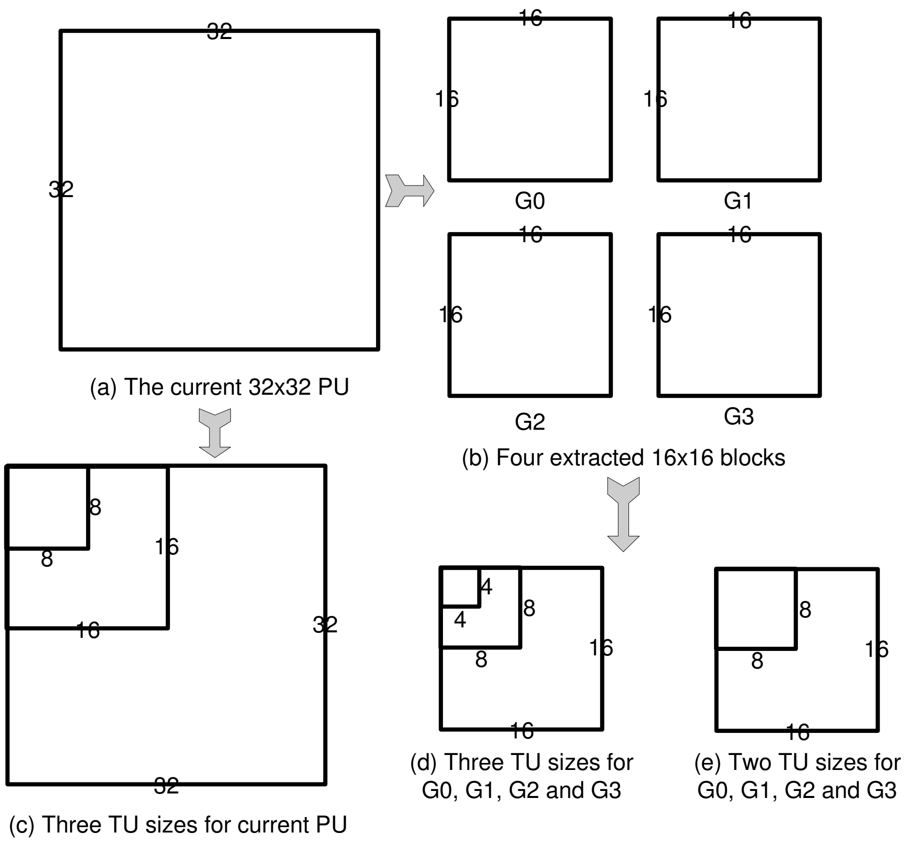


Figure 7.1. Structure of HIP for 32x32 PU and its corresponding TU sizes

7.2.2 To reduce coding complexity of an encoder based on HEVC

When an encoder has sufficient capability to compress video material with high or ultrahigh definition resolution, developing fast algorithms to reduce the coding complexity of the encoder with negligible loss in coding efficiency becomes increasingly important for real time applications. So another important future research direction is to reduce the coding complexity of an encoder based on the HEVC standard. In HEVC, the basic block is called LCU (the Largest Coding Unit) which can be as large as 64x64. Each video frame is firstly divided into many non-overlapped LCUs. Each LCU is then recursively split into smaller CUs forming a quad tree structure. The CU can be further split into PUs and TUs in turn. In order to find the optimal solution, the encoder needs to exhaust all combinations of CU, PU and TU. This is a very time-consuming process. To develop some fast algorithms while keeping similar rate distortion performance is imperative for the practicability of HEVC. In Chapters 3 and 4, early detection algorithms of ZQDCT coefficients and fast intra mode decision algorithm were shown to speed up the encoding process of an encoder based on H.264/AVC. We believe that they should also be effective in HEVC.

In HEVC, four different TU sizes ranging from 4x4 to 32x32 are supported to adapt to the varying space-frequency characteristics of the residual signal. Note that the maximum admissible depth is three in the current design of HM. The partitioning of a given CU (for inter coding) or PU (for intra coding) into TUs is done based on the RQT technique. With the maximum depth of three, the overall coding efficiency can be increased. However, the RQT based transform coding is a very complex process. Fortunately, according to the analysis and statistics of works on quantized DCT coefficients, many DCT coefficients become zero after quantization. In the case of low bitrate coding, even the whole residual blocks could be zero after quantization. We call these blocks as all-zero-blocks (AZBs). Furthermore, it is very likely that all the four subblocks at depth n+1 are AZBs when an AZB is judged at depth n. This property is called Zero-Block Inheritance (ZBI) in [40]. By using the proposed algorithms in Chapter 3, the AZB at depth n can be detected before taking the RQT process, and the corresponding DCT, Q, IQ and IDCT at depth n and the remaining higher depths can be eliminated. Consequently, considerable computation can be saved and the encoding time can be reduced for the encoder based on HM. Besides the detection of AZBs, parts of the ZQDCT coefficients can also be detected by using the proposed algorithm in Chapter 3. This also allows to make contributions to saving encoding time of an encoder based on HM.

Note that as many as 35 intra prediction modes are supported in the current HEVC to provide a more accurate prediction. In order to reduce the intra coding complexity, three-stage approach is adopted in HM [35] to select the best mode and the optimal RQT structure. As we have talked about that the RQT process could be speeded up by extending our proposed algorithms in Chapter 3. While the mode decision process can also be accelerated by extending our proposed algorithm in Chapter 4. The variance based technique has the potential to determine the PU size. And the candidate modes could then be selected based on our directional difference measurement. Moreover, the early termination strategy could also be extended into HM for RDO process. Furthermore, some mature algorithms, such as fast inter mode decision[104-107], fast motion estimation[108-112] and rate distortion estimation [113-118], proposed for H.264/AVC, could also be extended into HM. All these techniques could save much more encoding time and make the encoder more practical. This would be a fruitful research direction in future.

7.2.3 To compress 3D or Multiview videos using new techniques in HEVC

Note that 3D video technique has achieved a significant interest. Not only 3D video but also free view-point video technique is currently becoming a hot research topic. For these videos, storage or transmission is the major problem due to a huge amount of data required to represent them. The JVT has standardized an extension of H.264/AVC standard as Multiview Video Coding (MVC) to encode multiview videos [120]. The 3D video is a special case of multiview videos. Note that high compression capability has been achieved with the introduction of the coming HEVC standard. Hence, it is naturally selected as a research platform to compress 3D video or free view-point video as another direction of our research in future. In the MVC of H.264/AVC standard, not only inter-frame prediction but also inter-view prediction is available to compress the multiview videos. Inter-frame prediction is related to motion estimation, while inter-view prediction involves in disparity estimation. In our future research work, we would focus on developing some new algorithms to improve the prediction accuracy of inter-frame prediction and inter-view prediction, and thereby achieve a higher compression capability based on the HEVC standard. We would also investigate some techniques to accelerate motion estimation and disparity estimation processes.

Appendix

A.1 Derivation of p.d.f. of random variable M

Since M=|X|, and X has a Laplace distribution as shown in (3.8), the cumulative distribution function (cdf) of random variable M can be expressed as

$$F(M < m) = F(|X| < m) = F(-m < X < m) = \int_{-m}^{m} f_X(x)dx \tag{3.54}$$

Based on (3.54), the p.d.f. of M can be calculated as

$$f_M(m) = \begin{cases} f_X(m)(m)' - f_X(-m)(-m)' = \lambda_0 \, \mathrm{e}^{-\lambda_0 m}, & m \geq 0 \\ 0, & m < 0 \end{cases} \tag{3.55}$$

Equation (3.55) means that the random variable M follows an exponential distribution with parameter $1/\lambda_0$, labeled as M~exp($1/\lambda_0$).

A.2 Derivation of p.d.f. of random variable S

Let $Z=M_1+M_2$. Since M_1 and M_2 are independent variables with identical distribution (iid) as in (3.10), we have

$$\begin{aligned} f_Z(z) &= \int_{-\infty}^{\infty} f_{M_1}(m_1) f_{M_2}(z-m_1)\,dm_1 \\ &= \int_0^z \lambda_0 \, \mathrm{e}^{-\lambda_0 m_1} \lambda_0 \, \mathrm{e}^{-\lambda_0 (z-m_1)}\,dm_1 \\ &= \lambda_0^2 \, \mathrm{e}^{-\lambda_0 z} \int_0^z dm_1 = \lambda_0^2 z \, \mathrm{e}^{-\lambda_0 z} \end{aligned} \tag{3.56}$$

Hence, Z~Γ(2,$1/\lambda_0$).

Similarly, let $R=Z_1+Z_2$, and we then have

$$\begin{aligned} f_R(r) &= \int_{-\infty}^{\infty} f_{Z_1}(z_1) f_{Z_2}(r-z_1)\,dz_1 \\ &= \int_0^r \lambda_0^2 z_1 \, \mathrm{e}^{-\lambda_0 z_1} \lambda_0^2 (r-z_1) \, \mathrm{e}^{-\lambda_0 (r-z_1)}\,dz_1 = \lambda_0^4 r^3 \, \mathrm{e}^{-\lambda_0 r} / 3! \end{aligned}, \tag{3.57}$$

where 3!=3x2x1. Hence, R~Γ(4,1/λ_0), and so on.

Since $S = SAD = \sum_{i=0}^{15} |X_i| = \sum_{i=0}^{15} M_i$, which can be considered as the summation of 16 iid random variables. Hence, the random variable S follows a Gamma distribution, which is labeled as S~Γ(16,1/λ_0). The p.d.f. of it can be written as (3.12).

A.3 Derivation of T_{uv}

Substitute (3.30) into (3.31), and it gives (3.58) as follows.

$$\begin{aligned} J_1(T_{uv}) &= \frac{1}{N_1}\int_{T_{uv}^0}^{T_{uv}}\int_{Qstep}^{K_{uv}\cdot s+f\cdot Qstep} \frac{\lambda_0^{16}(s+s_0)^{15}}{15!} \mathrm{e}^{-\lambda_0(s+s_0)} \times \frac{3\sqrt{2}}{K_{uv}\cdot s}\mathrm{e}^{-\frac{3\sqrt{2}}{K_{uv}\cdot s}(g_{uv}-f\cdot Qstep)} dg_{uv}ds \\ &= \frac{\lambda_0^{16}}{N_1\cdot 15!}\int_{T_{uv}^0}^{T_{uv}} (s+s_0)^{15}\mathrm{e}^{-\lambda_0(s+s_0)}\mathrm{e}^{\frac{3\sqrt{2}}{K_{uv}\cdot s}f\cdot Qstep}\left(\int_{Qstep}^{K_{uv}\cdot s+f\cdot Qstep}\frac{3\sqrt{2}}{K_{uv}\cdot s}\mathrm{e}^{-\frac{3\sqrt{2}}{K_{uv}\cdot s}g_{uv}}dg_{uv}\right)ds \\ &= \frac{-\lambda_0^{16}}{N_1\cdot 15!}\int_{T_{uv}^0}^{T_{uv}} (s+s_0)^{15}\mathrm{e}^{-\lambda_0(s+s_0)}\left(\mathrm{e}^{-3\sqrt{2}}-\mathrm{e}^{\frac{3\sqrt{2}(f-1)\cdot Qstep}{K_{uv}\cdot s}}\right)ds \end{aligned} \tag{3.58}$$

Hence,

$$\begin{aligned} \frac{dJ_1(T_{uv})}{dT_{uv}} &= \frac{d}{dT_{uv}}\left[\frac{-\lambda_0^{16}}{N_1\cdot 15!}\int_{T_{uv}^0}^{T_{uv}} (s+s_0)^{15}\mathrm{e}^{-\lambda_0(s+s_0)}\left(\mathrm{e}^{-3\sqrt{2}}-\mathrm{e}^{\frac{3\sqrt{2}(f-1)\cdot Qstep}{K_{uv}\cdot s}}\right)ds\right] \\ &= \frac{-\lambda_0^{16}}{N_1\cdot 15!}(T_{uv}+s_0)^{15}\mathrm{e}^{-\lambda_0(T_{uv}+s_0)}\left(\mathrm{e}^{-3\sqrt{2}}-\mathrm{e}^{\frac{3\sqrt{2}(f-1)\cdot Qstep}{K_{uv}\cdot T_{uv}}}\right) \end{aligned} \tag{3.59}$$

Similarly, $dJ_2\,(T_{uv})/dT_{uv}$ can be calculated as follows:

$$\frac{dJ_2(T_{uv})}{dT_{uv}} = \frac{\lambda_0^{16}}{N_2\cdot 15!}(T_{uv}+s_0)^{15}\mathrm{e}^{-\lambda_0(T_{uv}+s_0)}\left(\mathrm{e}^{\frac{3\sqrt{2}(f-1)\cdot Qstep}{K_{uv}\cdot T_{uv}}}-1\right) \tag{3.60}$$

Substitute (3.59) and (3.60) into (3.38), it gives

$$\frac{1}{N_1}\left(\mathrm{e}^{-3\sqrt{2}}-\mathrm{e}^{\frac{3\sqrt{2}(f-1)\cdot Qstep}{K_{uv}\cdot T_{uv}}}\right)-\frac{\lambda}{N_2}\left(\mathrm{e}^{\frac{3\sqrt{2}(f-1)\cdot Qstep}{K_{uv}\cdot T_{uv}}}-1\right)=0 \tag{3.61}$$

As a result, (3.39) can be obtained from (3.61).

References

[1] Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, "Draft ITU-T Recommendation and Final Draft international Standard of Joint Video Specification (ITU-T Rec. H.264| ISO/ IEC 14496-10 AVC)", JVT-G050, Mar. 2003.

[2] T. Wiegand, G.J. Sullivan, G. Bjontegaard and A. Luthra, "Overview of the H.264/AVC Video Coding Standard", IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, no. 7. pp. 560-576. July 2003.

[3] I.E.G. Richardson, H.264 and MPEG-4 Video Compression: Video Coding for Next-Generation Multimedia, England, New York:Wiley, 2003.

[4] G.J. Sullivan, P. Topiwala, and A. Luthra, "The H.264/AVC Advanced Video Coding Standard: Overview and Introduction to the Fidelity Range Extensions," Proceedings, SPIE Conference on Applications ofDigital Image Processing XXVII, Aug. 2004.

[5] JVT H.264/AVC Reference Software, http://iphome.hhi.de/suehring/tml/download/.

[6] Y.H. Moon and G.Y. Kim, "An Improved Early Detection Algorithm for All-Zero Blocks in H.264 Video Encoding," IEEE Transactions on Circuits and Systems for Video Technology, Vol.15, No.8, pp.1053-1057, Aug. 2005.

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

[8] W.B. Pennebaker and J.L. Mitchell, JPEG: Still Image Data Compression Standard, Van Nostrand Reinhold, New York, 1992.

[9] X. Li and M.T. Orchard, "Edge-based prediction for lossless compression of Natural Images," IEEE Transactions on Image Processing, vol.10, no.6, pp.813-817, June 2001.

[10] K. Ugure, K. Andersson, A. Fuldseth, G. Bjontegaard, L.P. Endresen, J. Lainema, A. Hallapuro, J. Ridge, D. Rusanovskyy, C.X. Zhang, A. Norkin, C. Priddle, T. Rusert, J. Samuelsson, R. Sjoberg, Z.F. Wu, "High Performance, Low Complexity Video Coding and the Emerging HEVC Standard," IEEE Transactions on Circuits and Systems for Video Technology, Vol.20, No.12, pp.1688-1697, Dec. 2010.

[11] B. Bross, W.J. Han, G. J. Sullivan, J.R. Ohm and T. Wiegand, "High Efficiency Video Coding (HEVC) text specification draft 6", JCTVC-H1003, San Jose, Feb. 2012.

[12] X. Zhou, Z.H. Yu, and S.Y. Yu, "Method for detection all-zero DCT coefficients ahead of discrete cosine transform," Electron. Lett., vol.34, no.19, pp.1839-1840, Sep. 1998.

[13] L.A. Sousa, "General method for eliminating redundant computations in video coding," Electron. Lett., vol.36, no.4, pp.306-307, Feb. 2000.

[14] Y.H. Moon and G.Y. Kim, "An Improved Early Detection Algorithm for All-Zero Blocks in H.264 Video Encoding," IEEE Transactions on Circuits and Systems for Video Technology, vol.15, no.8, pp.1053-1057, Aug. 2005.

[15] J. Li, J. Takala, M. Gabbouj. and C.X. He, "A detection algorithm for zero-quantized DCT coefficients in JPEG," Proceedings, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'2008), pp.1189-1192, Mar. 2008, Las Vegas, USA.

[16] Z. Xie, Y. Liu, J. Liu and T. Yang, "A General Method for Detecting All-Zero Blocks Prior to DCT and Quantization," IEEE Transactions on Circuits and Systems for Video Technology, vol.17, no.2, pp.237-241, Feb. 2007.

[17] H. Wang and S. Kwong, "Hybrid Model to Detect Zero Quantized DCT Coefficients in H.264," IEEE Transactions on Multimedia, vol.9, no.4, pp.728-735, Jun. 2007.

[18] M. Pao and M. T. Sun, "Modeling DCT Coefficients for Fast Video Encoding," IEEE Transactions on Circuits and Systems for Video Technology, vol.9, no.4, pp.608-616, Jun. 1999.

[19] M.J. Zhang, T. Zhou and W. Wang, "Adaptive Method for Early Detecting Zero Quantized DCT Coefficients in H.264/AVC Video Encoding," IEEE Transactions on Circuits and Systems for Video Technology, vol.19, no.1, pp.103-107, Jan. 2009.

[20] Y. Zhao and Y.L. Yuan, "All-Zero Block Detecting Algorithm based on Look-Up Threshold Tables in H.264," Proceedings, International Conference on Digital Content, Multimedia Technology and its Applications (IDC'2010), pp.183-187, Aug. 2010, Seoul, Korea.

[21] W.Y. Chiu, Y.M. Lee and Y.Y. Lin, "Efficient zero-block mode decision algorithm for high bit-rate coding in H.264/AVC," Proceedings, IEEE International Symposium on Circuits and Systems (ISCAS'2010), pp.4189-4192, May 2010, Paris, France.

[22] P. Feng, X. Lin, S. Rahardja, K.P. Lim, Z.G. Li, D,J. Wu, and S. Wu. "Fast Mode Decision Algorithm for Intra prediction in H.264/AVC Video Coding," IEEE Transactions on Circuits and Systems for Video Technology, vol.15, no.7, pp.813-822. July 2005.

[23] J.C. Wang, J.F. Wang, J.F. Yang and J.T. Chen, "A fast mode decision algorithm and its VLSI design for H.264/AVC intra-prediction," IEEE Transactions on Circuits and Systems for Video Technology, vol. 17, no.10, pp.1414-1422, Oct. 2007.

[24] A.C. Tsai, J.F. Wang, J.F. Yang and W.G. Lin, "Effective Subblock-Based and Pixel-Based Fast Direction Detections for H.264 Intra Prediction," IEEE Transactions on Circuits and System for Video Technique, vol.18. no.7, pp.975-981, July 2008.

[25] Y.H. Huang, T.S. Ou and H.H. Chen, "Fast Decision of Block Size, Prediction Mode, and Intra Block for H.264 Intra Prediction," IEEE Transactions on Circuits and System for Video Technique, vol.20. no.8, pp.975-981, Aug. 2010.

[26] Y.L. Lee, K.H. Han and G.J. Sullivan, "Improved lossless intra coding for H.264/MPEG-4 AVC," IEEE Transaction on Image Processing, vol.15, no.9, pp.2610-2615, Sep. 2006.

[27] S.T. Wei, S.R. Shen, B.D. Liu and J.F. Yang, "Lossless image and video coding based on H.264/AVC intra predictions with simplified interpolations," Proceeding, IEEE International Conference on Image Processing (ICIP'2009), pp.633-636, Nov. 2009, Cairo, Egypt.

[28] J.R. Ding, J.Y. Chan, F.C. Yang and J.F. Yang, "Two-layer and adaptive entropy coding algorithms for H.264-based lossless image coding," Proceedings, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'2008), pp.1369-1372, Apr. 2008, Las Vegas, USA.

[29] L. Song, Z.Y. Luo and C. Xiong, "Improving Lossless Intra Coding of H.264/AVC by Pixel-Wise Spatial Interleave Prediction," IEEE Transactions on Circuits and Systems for Video Technology, vol.21, no.12, pp.1924-1928, Dec. 2011.

[30] L.L. Wang and W.C. Siu, "Improved Lossless Coding Algorithm in H.264/AVC based on Hierarchical Intra Prediction and Coding Mode Selection," Journal of Electronic Imaging 20(4), 043001 (Oct.-Dec. 2011).

[31] J. Lainema and K. Ugur, "Angular Intra Prediction in High Efficiency Video Coding (HEVC)," Proceedings, Multimedia Signal Processing (MMSP'2011), Oct. 2011, Tampere, Finland.

[32] H.L. Tan, C.H. Yeo, Y.H. Tan and Z.G. Li, "Non CE6: Unification of the number of intra prediction modes," JCTVC-H0166, San Jose, USA, Feb. 2012.

[33] C. Lai, L. Liu and J. Zheng, "Non CE6: Unify the intra mode number for all PU size," JCTVC-H0342, San Jose, USA, Feb. 2012.

[34] Y. Piao, J. Min and J. Chen, "Encoder improvement of unified intra prediction," JCTVC-C207, Guangzhou, Oct. 2010.

[35] https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/

[36] D. Marpe, H. Schwarz, S. Bosse, B. Bross, P. Helle, T. Hinz, H. Kirchhoffer, H. Lakshman, T. Nguyen, S. Oudin, M. Siekmann, K. Suhring, M. Winken and T. Wiegand, "Video Compression Using Nested Quadtree Structure, Leaf Merging, and Improved Techniques for Motion Representation and Entropy

Coding," IEEE Transactions on Circuits and Systems for Video Technology, vol.20, no.12, pp.1676-1687, Dec. 2010.

[37] M. Winken, P. Helle, D. Marpe, H. Schwarz and T. Wiegand, "Transform Coding in the HEVC Model," Proceedings, International Conference on Image Processing (ICIP'2011), pp.3693-3696, Sep. 2011, Brussels, Belgium.

[38] J. Kim, J. Yang, H. Lee and B. Jeon, "Fast Intra Mode Decision of HEVC based on Hierarchical Structure," Proceedings, IEEE International Conference on Information, Communications and Signal Processing (ICICS'2011), Dec. 2011, Singapore.

[39] L. Zhao, L. Zhang, S. Ma and D. Zhao, "Fast Mode Decision Algorithm for Intra Prediction in HEVC," Proceedings, Visual Communications and Image Processing (VCIP'2011), Nov. 2011, Tainan City, Taiwan.

[40] S.W. Teng, H.M. Hang and Y.F. Chen, "Fast Mode Decision Algorithm for Residual Quadtree Coding in HEV," Proceedings, Visual Communications and Image Processing (VCIP'2011), Nov. 2011, Tainan City, Taiwan.

[41] G.V. Wallendael, S.V. Leuven, J.D. Cock, P. Lambert, R.V. Walle, J. Barbarien and A. Munteanu, "Improved Intra Mode Signaling for HEVC," Proceedings, IEEE International Conference on Multimedia and Expo (ICME'2011), July. 2011, Barcelona, Spain.

[42] M.J. Weinberger, "The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS," IEEE Transaction on Image Processing, vol.9, no.8, pp.1309-1324, Aug. 2000.

[43] Motion JPEG2000 Final Committee Draft, ISO/IEC JTC 1/SC29/WG1, Document N2117, Mar. 2001.

[44] F. Bossen, "Common test conditions and software reference configurations," JCTVC-G1200, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T VCEG and ISO/IEC MPEG, Geneva, CH, Feb. 2011.

[45] T. Zhou, M. Zhang and Z. Xiong, "Generalised Gaussian distribution to early detect zero quantised discrete cosine transform coefficients in H.264/AVC video encoding," IET Image Process., Vol.4, Iss.6, pp.473-485, 2010.

[46] A.K. Jain, Fundamentals of Digital Image Processing. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1989.

[47] W.K. Pratt, Digital Image Processing. New York: Wiley, 1978.

[48] E.Y. Lam and J.W. Goodman, "A Mathematical Analysis of the DCT Coefficient Distributions for Images," IEEE Trans. on Image Processing, vol.9, no.10, Oct. 2000.

[49] H. Wang, S. Kwong and C.W. Kok, "Efficient Prediction Algorithm of Integer DCT Coefficients for H.264/AVC Optimization," IEEE Transactions on Circuits and Systems for Video Technology, vol.16, no.4, pp.547-552, Apr. 2006.

[50] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," presented at the 13th VCEG-M33 Meeting, Austin, Texas, USA, 2-4 April, 2001.

[51] I. Choi, J. Lee and B. Jeon, "Fast Coding Mode Selection With Rate-Distortion Optimization for MPEG-4 Part-10 AVC/H.264," IEEE Transactions on Circuits and System for Video Technique, vol. 16, no. 12, pp.1557-1561, Dec. 2006.

[52] K.C. Hui, W.C. Siu and Y.L. Chan, "New Adaptive Partial Distortion Search using Clustered Pixel Matching Error Characteristic," IEEE Transactions on Image Processing, vol.14, no.5, pp.597-607, May 2005.

[53] Y.L. Chan and W.C. Siu, "New Adaptive Pixel Decimation for Block Motion Vector Estimation," IEEE Transactions on Circuits and Systems for Video Technology, pp.113-118, vol.6, no.1, Feb. 1996.

[54] A. Paul, J. Wu, J.F. Yang and J. Jeong, "Gradient-based edge detection for motion estimation in H.264/AVC," IET Image Processing, Vol.5, Iss.4, pp.323-327, 2011.

[55] R. Reininger and J. Gibson, "Distributions of the two-dimensional DCT coefficients for images," IEEE Trans. Commun., vol. COM-31, pp. 835-839, June 1983.

[56] D.S. Turaga, Y. Chen, and J. Caviedes, “No reference PSNR estimation for compressed pictures,” Signal Processing: Image Communication, vol. 19, pp.173-184, Feb. 2004.

[57] M.A. Robertson and R.L. Stevenson, “DCT Quantization Noise in Compression Images,” IEEE Transactions on Circuits and Systems for Video Technology, vol.15, no.1, pp.27-38, Jan. 2005.

[58] SYRIS Technology Corp., “Technical Document About FAR, FRR and EER,” Ver.1.0, 2004.

[59] G. Sullivan, “Recommended Simulation Common Conditions for Coding Efficiency Experiments Revision 1,” 31st VCEG-AE010R1 meeting, Marrakech, Morocco, 13-19 January, 2007.

[60] S. Milani, “Fast H.264/AVC FRExt Intra Coding Using Belief Propagation,” IEEE Transactions on Image Processing, vol.20, no.1, pp.121-131, Jan. 2011.

[61] L.L. Wang and W.C. Siu, “H.264 fast intra mode selection algorithm based on direction difference measure in the pixel domain,” Proceedings, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP’2009), pp.1037-1040, Apr. 2009.

[62] A. Yu, G. Martin and H. Park, “Fast inter-mode selection in the H.264/AVC standard using a hierarchical decision process,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 18, no. 2, pp. 186-195, Feb. 2008.

[63] J. Hhhm and C. Kyung, “Efficient CABAC Rate Estimation for H.264/AVC Mode Decision,” IEEE Transactions on Circuits and System for Video Technique, vol. 20, no. 2, pp.310-316, Feb. 2010.

[64] F. Pan, X. Lin, S. Rahardja, K.P. Lim, and Z.G. Li, “A directional field based fast intra mode decision algorithm for H.264 video coding,” Proceedings, IEEE International Conference on Multimedia and Expo (ICME’2004), pp. 1147-1150, Jun. 2004, Taipei, Taiwan.

[65] A.C. Yu, N.K. Ngi and G.R. Martin, “Efficient intra- and inter-mode selection algorithms for H.264/AVC,” Journal of Visual Communication and Image Representation, vol. 17, Issue 2, pp.322-344, Apr. 2006.

[66] Y. Zhang, W.C. Siu and T.Z. Shen, "Yet a Faster Motion Estimation Algorithm with Directional Search Strategies," Proceedings, 15th International Conference on Digital Signal Processing (DSP'2007), pp.475-478, July 2007, Cardiff, UK.

[67] J. Heo, S.H. Kim and Y.S. Ho, "Improved CAVLC for H.264/AVC lossless intra coding," IEEE Transactions on Circuits and Systems for Video Technology, vol. 20, no. 2, pp.213-222, Feb. 2009.

[68] Y. Zhang, W.C. Siu, and T.Z. Shen, "Comprehensive scheme for subpixel variable block-size motion estimation," Journal of Electronic Imaging 20(1), Jan.-Mar. 2011.

[69] V.R. Algazi, G.E. Ford and R. Potharlanka, "Directional interpolation of images based on visual properties and rank order filtering," Proceedings, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'1991), pp.3005-3008, May 1991, Ontario, Canada.

[70] S.W. Lee and J.K. Paik, "Image interpolation using adaptive fast B-spline filtering," Proceedings, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'1993), pp.177-180, Apr. 1993, Minnesota, USA.

[71] J.E.A. Jr, "Interactions between color plane interpolation and other image processing functions in electronic photography," Proc. SPIE, vol.2416, pp. 144-151, Mar. 1995.

[72] S. Carrato, G. Ramponi and S. Marsi, "A simple edge-sensitive image interpolation filter," Proceedings, IEEE International Conference on Image Processing (ICIP'1996), pp.711-714, Sep. 1996, Lausanne, Switzerland.

[73] Y. Vatis, B. Edler, D.T. Nguyen and J. Ostermann, "Two-dimensional non-separable Adaptive Wiener Interpolation Filter for H.264/AVC," ITU-T SGI 6/Q.6 Doc. VCEG-Z17, Busan, South Korea, April 2005.

[74] Y. Vatis and J. Ostermann, "Adaptive Interpolation Filter for H.264/AVC," IEEE Transactions on Circuits and Systems for Video Technology, vol.19, no.2, pp.179-192, Feb. 2009.

[75] H.M. Hang, W.H. Peng, C.H. Chan, and C.C. Chen, “Towards the next video standard: high efficiency video coding,” Asia-Pacific Signal and Information Processing Association Annual Summit and Conf., pp.609-618, Dec. 2010.

[76] P. R. Hill, T. K. Chiew, D. R. Bull, and C. N. Canagarajah, “Interpolation free subpixel accuracy motion estimation,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 16, no. 12, pp. 1519–1526, Dec. 2006.

[77] T.K. Chiew, J.C. How, D.R. Bull and C.N. Canagarajah, “Interpolation free subpixel refinement for block-based motion estimation,” Proceedings, The SPIE Visual Communications and Image Processing, vol.5308, pp.1261-1269, 2004.

[78] T. Wedi, “Adaptive interpolation filters and high-resolution displacements for video coding,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 16, no. 4, pp. 484–491, Apr. 2006.

[79] X. Li and M.T. Orchard, “New Edge-Directed Interpolation,” IEEE Transactions on Image Processing, vol.10, no.10, pp.1521-1527, Oct. 2001.

[80] X.J. Zhang and X.L. Wu, “Image Interpolation by Adaptive 2-D Autoregressive Modeling and Soft-Decision Estimation,” IEEE Transactions on Image Processing, vol.17, no.6, pp.887-896, June 2008.

[81] HP Laboratories. JPEG-LS Reference Encoder [Online]. Available: http://www.hpl.hp.com/loco/locodown.htm

[82] S. Wittmann and T. Wedi, “Separable Adaptive Interpolation Filter,” ITU-T SG 16/Q.6 Doc. T05-SG16-C0219, Geneva, Switzerland, Jun. 2007.

[83] D. Rusanovskyy, K. Ugur and J. Lainema, “Adaptive Interpolation with Directional Filters,” ITU-T SG 16/Q.6 Doc. VCEG-AG21, Shenzhen, China, Oct. 2007.

[84] X. Li, N. Oertel, A. Hutter and A. Kaup, “Laplace Distribution Based Lagrangian Rate Distortion Optimization for Hybrid Video Coding,” IEEE Transactions on Circuits and Systems for Video Technology, vol.19, no.2, pp.193-205, Feb. 2009.

[85] K.Y. Liao, J.F. Yang and M.T. Sun, “Rate-Distortion Cost Estimation for H.264/AVC,” IEEE Transactions on Circuits and Systems for Video Technology, vol.20, no.1, pp.38-49, Jan. 2010.

[86] Y. Chan, Y.K. Wang, K. Ugur, M. Hannuksela, J. Lainema and M. Gabbouj, “The emerging MVC standard for 3D video services,” EURASIP J. Adv. Signal Process., vol. 2009, no.1, Jan. 2009, article ID 786015.

[87] S. Matsuo, S. Takamura, K. Kamikura and Y. Yashima, “Extension of Intra Prediction Using Multiple Reference Lines,” ITU-T Q.6/SG16 VCEG, VCEG-AF05, April 2007.

[88] D.C. Garcia and R.L. Queiroz, “Least-Squares Directional Intra Prediction in H.264/AVC,” IEEE Signal Processing Letters, vol.17, no.10, Oct. 2010.

[89] A. Masmoudi, W. Puech and M.S. Bouhlel, “Efficient adaptive arithmetic coding based on updated probability distribution for lossless image compression,” Journal of Electronic Imaging 19(2), Apr.-Jun. 2010.

[90] K. Zhang, S. Lei and W. Gao, “Enhanced Intra Prediction and Transform Video Coding,” Proceedings, IEEE International Conference on Image Processing (ICIP’2010), pp.3381-3384, Sep. 2010, Hong Kong.

[91] C.W. Wong and W.C. Siu, “Transform Kernel Selection Strategy for the H.264/AVC and Future Video Coding Standards,” IEEE Transactions on Circuits and Systems for Video Technology, vol.21 no.11, pp.1631-1645, Nov. 2011.

[92] J.N. Han, A. Saxena, V. Melkote and K. Rose, “Jointly Optimized Spatial Prediction and Block Transform for Video and Image Coding,” IEEE Transactions on Image Processing, vol.21, no.4, pp.1874-1884, April 2012.

[93] G.J. Sullivan, J.R. Ohm, F. Bossen and T. Wiegand, “JCT-VC AHG report: HM subjective quality investigation,” JCTVC-H0022, Feb. 2012.

[94] S. Ma and C.C.J. Kuo, “High-definition video coding with supermacroblocks,” Proc. SPIE, vol. 6508, part 1, p. 650816, Jan. 2007.

[95] A. Fuldseth, M. Horowitz, S. Xu, A. Segall and M. Zhou, “Tiles,” JCTVC-F335, July 2011.

[96] K. Misra, J. Zhao and A. Segall, “Lightweight slicing for entropy coding,” JCTVC-D070, Jan. 2011.

[97] C. Gordon, F. Henry and S. Pateux, “Wavefront parallel processing for HEVC encoding and decoding,” JCTVC-F274, July 2011.

[98] J. Chen and T. Lee, “Planar intra prediction improvement,” JCTVC-F483, July 2011.

[99] J. Chen, V. Seregin, W.J. Han, J. Kim and J. Moon, “CE6.a.4: Chroma intra prediction by reconstructed luma samples,” JCTVCE266, Mar. 2011.

[100] C.M. Fu, C.Y. Chen, Y.W. Huang and S. Lei, “Sample adaptive offset for HEVC,” Proceedings, IEEE International Workshop on Multimedia Signal Processing (MMSP’2011), Oct. 2011, Hangzhou, China.

[101] E. Alshina, A. Alshin, J.H. Park, J. Lou and K. Minoo, “CE3: 7 taps interpolation filters for quarter pel position MC from Samsung and motorola mobility,” JCTVC-G778, Geneva, Nov. 2011.

[102] J. Han, A. Saxena and K. Rose, “Towards jointly optimal spatial prediction and adaptive transform in video/image coding,” Proceedings, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP’2010), pp. 726-729, Mar. 2010, Texas, USA.

[103] K. Choi and E.S. Jang, “Early TU decision method for fast video encoding in high efficiency video coding,” Electronics Letters, vol.48, no.12, Jun. 2012.

[104] C. Grecos and M. Yang, “Fast mode prediction for the baseline and main profiles in the H.264 video coding standard,” IEEE Transactions on Multimedia, vol. 8, Issue 6, pp. 1125-1134, Dec. 2006.

[105] M.E. Eduardo, J.M. Amaya and D.M. Fernando, “An Adaptive Algorithm for Fast Inter Mode Decision in the H.264/AVC Video Coding Standard,” IEEE Transactions on Consumer Electronics, vol. 56, no.2, pp. 826-834, May 2010.

[106] S.W. Jung, S.J. Baek, C.S. Park and S.J. Ko, “Fast Mode Decision Using All-Zero Block Detection for Fidelity and Spatial Scalable Video Coding,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 20, no. 2, pp. 201–206, Feb. 2010.

[107] M.E. Eduardo, J.M. Amaya and A.P. Miguel, “A Two-Level Classification-Based Approach to Inter Mode Decision in H.264/AVC,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 21, no. 11, pp. 1719–1732, Nov. 2011.

[108] T.Y. Kuo and C.H. Chan, “Fast variable block size motion estimation for H.264 using likelihood and correlation of motion field,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 16, no. 10, pp. 1185–1195, Oct. 2006.

[109] C.H. Fu, Y.L. Chan and W.C. Siu, “Fast Motion Estimation and Mode Decision for H.264 Reverse Transcoding,” Electronic Letters, vol.42, no.24, pp.1385-1386, Nov. 2006.

[110] M.Y. Chiu and W.C. Siu, “Computationally-Scalable Motion Estimation Algorithm for H.264/AVC Video Coding,” IEEE Transactions on Consumer Electronics, vol.56, no.2, pp.895-903, May, 2010.

[111] Y. Zhang, W.C. Siu and T.Z. Shen, “A Comprehensive Scheme for Sub-pixel Variable Block Size Motion Estimation,” Journal of Electronic Imaging, pp.1-20, Vol.20(1), No.013014, Jan.-Mar. 2011.

[112] Y. Ismail, J.B. Mcneely, M. Shaaban, H. Mahmoud and M.A. Bayoumi, “Fast Mostion Estimation System Using Dynamic Models for H.264/AVC Video Coding,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 1, pp. 28-42, Jan. 2012.

[113] S.C. Chang, J.F. Yang, C.F. Lee and J.N. Hwang, “A novel rate predictor based on quantized DCT indices and its rate control mechanism,” Signal Process.: Image Commun., vol. 18, pp. 427-441, Jul. 2003.

[114] Q. Chen and Y. He, “A fast bits estimation method for rate-distortion optimization in H.264/AVC,” Proceedings, International Picture Coding Symposium (PCS’2004), pp.133-134, Dec. 2004, CA, USA.

[115] Y.K. Tu, J.F. Yang and M.T. Sun, “Efficient Rate-Distortion Estimation for H.264/AVC Coders,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 16, no. 5, pp. 28–42, May 2006.

[116] M.G. Sarwer and L.M. Po, "Fast Bit Rate Estimation for Mode Decision of H.264/AVC," IEEE Transactions on Circuits and Systems for Video Technology, vol. 17, no. 10, pp. 1402–1407, Oct. 2007.

[117] Y.K. Tu, J.F. Yang and M.T. Sun, "Rate-distortion modeling for efficient H.264/AVC encoding," IEEE Transactions on Circuits and Systems for Video Technology, vol. 17, no. 5, pp. 530–543, May 2007.

[118] J. Hahm, J. Kim and C.M. Kyung, "A fast CABAC rate estimation for H.264/AVC mode decision," Proceedings, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'2009), pp.929-932, Apr. 2009, Taipei, Taiwan.

[119] M.T. Pourazad, C. Doutre, M. Azimi and P. Nasiopoulos, "HEVC: The New Gold Standard for Video Compression: How Does HEVC Compare with H.264/AVC," IEEE Consumer Electronics Magazine, vol.1, pp.36-46, July 2012.

[120] A. Vetro, T. Wiegand and G.J. Sullivan, "Overview of the Stereo and Multiview Video Coding Extensions of the H.264/MPEG-4 AVC Standard," Proceedings of the IEEE, vol. 99, no.4, pp.626-642, Apr. 2011.