Waseda University Doctoral Dissertation

Research on Computational Complexity Optimization for Intra and Inter Prediction in Video Compression

Guifen TIAN

Graduate School of Information, Production and Systems
Waseda University

May 2013

Abstract

Due to the popularization of digital image capturing devices, video content is growing in volume every year. To compress the huge amount of data to much smaller capacity so that it can be manipulated with low cost for video conference, broadcasting, online video-on-demand and surveillance system, video coding technology has been developed since 1989. H.264/AVC is the latest international standard which provides high compression efficiency while introducing heavy computational load and high power consumption. As video content is also growing in resolution and 4k2k content becomes the mainstream, since 2010, a successor to H.264/AVC called High Efficiency Video Coding (HEVC) for next generation video coding is under standardization. HEVC can provide doubled compression efficiency compared to H.264/AVC, while tremendous computational cost is required which leads to longer encoding time and higher power consumption.

In both standards, the most effective while computationally intensive tool is Rate Distortion Optimization (RDO) based hybrid encoding, which reduces spatial and temporal redundancy by brutally trying a wide range of block types and prediction modes. In hybrid encoding, intra prediction reduces data redundancy in neighboring blocks, it is not that complexity consuming but affects throughput due to high data dependency; inter prediction estimates motions by finding a matching block in other several frames, consequently, large memory as well as high power consumption will be consumed. The problem becomes even critical in HEVC in that quadtree-structured coding block candidates for hybrid encoding have been increased by a factor of 2. Since video-related entertainment applications require real-time transmission and portable devices suffer from short battery life and limited hardware resources, it is desirable and urgent to develop optimization techniques for video encoders.

The target of this dissertation is to reduce the computational complexity in video encoder by half while keeping the compression performance in terms of both video quality and encoded bits, so that video data can be compressed efficiently within halt time or under lower power consumption.

Firstly, a mode decision scheme is proposed and implemented in to H.264/AVC, in which both prediction modes in intra coding and block types in hybrid prediction is reduced largely by means of texture analysis. Proposed texture analysis achieves better accuracy than previous works and the proposed mode decision scheme is proved to be capable of reducing computational complexity of H.264/AVC encoder by 52.5% averagely, which outperforms Liu's method [IEEE Trans. of CSVT 2009]

Secondly, a content adaptive hierarchical decision approach of variable coding blocks is proposed for compression 4k2k videos using HEVC encoder. In this approach, multiple features in temporal, spatial and transform domains are extracted for deciding the most suitable coding blocks and optimization can be achieved for all kinds of videos containing different characteristics in texture and motion. Comparison results with a proposal [JCTVC-F092 2011] adopted by HEVC demonstrates that higher complexity reduction is achieved by our proposed approach.

Thirdly, an ultra low complexity optimization scheme based on All-Zero Block (AZB) detection is brought forward to reduce power consumption for HEVC encoder. In this proposal, we derive a near-sufficient condition to detect variable-sized AZB in three video channels. Based on the condition, a technique to early terminate motion estimation and a pruning technique to constrain non-contributable prediction blocks are proposed. Experiments show that our scheme reduces the computational complexity of HEVC by up to 70.46% and 50.34% on average. The proposal also outperforms the latest method proposed by Shen [PCS 2012].

This dissertation consists of five chapters which are organized as follows.

Chapter 1 **[Introduction]** gives a brief introduction to the latest H.264/AVC standard and the emerging standard of HEVC. The most computationally intensive coding tool of RDO based mode decision and quadtree-structured hybrid predictions including residual encoding are discussed, followed by the motivation and contribution of this work.

Chapter 2 **[Mode Decision Scheme for H.264/AVC Based on Homogeneity Detection]** presents a fast mode decision algorithm to reduce computational complexity in hybrid prediction for H.264/AVC standard. Proposed algorithm consists of an edge direction detection method to constrain intra mode candidates and a novel homogeneity detection method for block type decision. The homogeneity detection method takes utilization of entropy theory and can select 16x16 or 4x4 block appropriately for intra coding; Meanwhile, either the large blocks in {16x16, 16x8, 8x16} or sub-blocks in {8x8, 8x4, 4x8, 4x4} are chosen for inter coding. Sufficient simulations demonstrate that consistent encoding gain is achieved for all videos with different motion and spatial features. Average Encoding complexity reduction is 52.5% with negligible loss in bitrate and PSNR, which outperforms Liu’s method [IEEE Trans. of CSVT 2009] by 12% more encoding time reduction and 0.55% better bitrate performance.

Chapter 3 **[Content Adaptive Hierarchical Decision of Variable Block Sizes in HEVC]** brings forward an adaptive scheme that exploits feature in temporal, spatial and transform domains to speed up the original quadtree-based prediction, targeting at

videos in resolution of HD and beyond. Before encoding starts, analysis on utilization ratio of each coding depth is performed to skip rarely adopted depths at frame level. Then, texture complexity (TC) measurement is applied to filter out none-contributable coding blocks for each largest coding unit (LCU). In this step, a dynamic threshold setting approach is proposed to make filtering adaptable to videos and coding parameters. Thirdly, during encoding process, sum of absolute quantized residual coefficient (SAQC) is used to prune useless coding blocks at CU level. Proposed scheme speeds up original HEVC encoder by a factor of up to 61.89% for 4kx2k video sequences and is superior to the proposal [JCTVC-F092 2011] adopted by HEVC in terms of complexity reduction of 16% with little bitrate distortion.

Chapter 4 **[All-Zero Block-Based Optimization for Quadtree-Structured Prediction and Residual Encoding]** presents a solution which takes full advantage of AZB for acceleration. A near-sufficient condition for detecting AZB is proposed first and for detected AZB, discrete cosine transform (DCT) and quantization is skipped in residual encoding. Using the condition, we also propose an early termination technique to reduce the complexity of motion estimation. More significantly, we present a two-dimensional pruning technique to constrain prediction units that contribute negligibly to rate-distortion performance. Experiments on videos with resolution ranging from 416×240 to 4k×2k show that the proposed scheme introduces little computation overhead and it is suitable for hardware implementation, it also accelerates HEVC encoder by up to 70.46% (50.34% on average), which outperforms a newest work [PCS 2012] with greater complexity reduction and 1.14% improved bitrate performance.

Chapter 5 **[Conclusion]** concludes the contribution of this dissertation.

Acknowledgment

It is not easy for me to come finally to this step writing an acknowledgment. This moment has been waited for a long time and it is so special to not only me but also to those who have always cared about, encouraged and helped me for the past years.

First and forever, I would like to appreciate my supervisor, Professor Satoshi Goto, for his consistent encouragement and generous support. It is under his inspiring and insightful supervision that I can grow up from a university graduate knew nothing about research to someone who can finish research tasks as scheduled and carry out research activities independently. Beyond research, what’s more is his devotion and hardworking spirits which influences me so profoundly that I will always remember his attitude towards the future research and life.

I would like to appreciate Professor Kimura and Professor Katto for their concise instructions through my research. Their valuable advices and comments are of great help to improve my presentation and the dissertation.

I would like to appreciate Professor Xin Jin. Every discussion with her is always inspiring. Moreover, I learned a lot of practical knowledge on how to organize and present research results to others in an understandable and acceptable way.

Thank all the members of Goto Lab. for their kindness. Special thanks are appreciated to Mr. Yichao Lu. In my most stressful time, he always encouraged me so much that I can concentrate on research and move forward.

Finally, I would like to appreciate my patents, brother and sister, for their love, understanding and fully support.

Without the precious encourage and help from the people above, I cannot accomplish this research. Thank you so much.

Contents

List of Tables

List of Figures

1 INTRODUCTION

1.1 Introduction to The Latest Video Compression Standards

Due to the popularization of digital image capturing devices, video content is growing in volume every year. To compress the huge amount of video data so that it can be manipulated with low cost for video conference, storage, online video-on-demand and surveillance system, video coding technology has been developed since 1989.

The most widely used international video encoding standard is known as ISO MPEG-x series [1] and the ITU-T H.26x series [2]. The MPEG (Moving Picture Coding Experts Group) series includes MPEG-1, MPEG-2, MPEG-4, for audio, image and video system. MPEG-1 was designed to produce reasonable quality images (352x240 pixels) and sound at low bit rates to fit the bandwidth of CD-ROM. MPEG-2 was designed to produce higher quality images at resolution of 720x486 at higher bit rates of 3-10 M bits per second for DVD, broadcast. MPEG-4 supports both low bit rates and high bitrates and is used in broadcasting, interactive and conversational environments.

The ITU (International Telecommunication Union) series includes H.261, H.263 and H.264 standards also for video compression established in 1990, 1995 and 2004, respectively. The H.261 is an international standard for ISDN picture phones and for video conferencing system, it supports images with QCIF and CIF formats at frame rate from 7.5 to 30 fps at a typical bit rate of 128 K bps. The H.263 [3] is a standard for picture phones over analog subscriber lines, it supports Sub-QCIF, QCIF and CIF formats below 10 fps at a typical bit rate of 20 Kbps and it is widely used as compression engine for internet video streaming.

The basic concept for video compression in all the above standards is the same that is block-oriented motion-compensation-based way to compress only residual data and then transform the residual to frequency domain followed by entropy encoding [57]. As more coding tools are gradually imported, basically, the later standards always outperform its predecessor.

1.1.1 Introduction to H.264/AVC

The latest international standard known as H.264 or MPEG-4 Part 10 Advanced Video Coding (AVC) [4] is developed jointly by Joint Video Team (JVT) of the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC MPEG in 2003. H.264/AVC significantly outperform previous H.263 by better compression efficiency with a range of features supporting high-quality, low bit-rate streaming video. In fact, compared to previous coding standards, H.264/AVC has shown significantly better coding performance in terms of peak signal-to-noise (PSNR) and visual quality at same bit rate [5]. The high coding performance enables H.264/AVC to be of great commercial value in applications like online video-on-demand, digital camera, digital media storage and broadcasting system, etc.

In technical viewpoint, above advancement of H.264 is due to a set of new features adopted such as Variable block sizes mode estimation (VBSME) at quarter pixel precise [6], multiple reference frames and multiple prediction modes for different block sizes used in intra coding [7]. Integer transform, in-loop de-blocking filter and content-based adaptive binary coding (CABAC) are also utilized to enhance coding performance. In addition, a computationally intensive tool called Rate Distortion Optimization (RDO) for selecting the optimal block size and prediction mode is employed to maximize coding quality and minimize data bits [8]. Figure 1.1 shows the video encoder diagram for H.264/AVC with intra and inter prediction forming a hybrid encoding sketch.

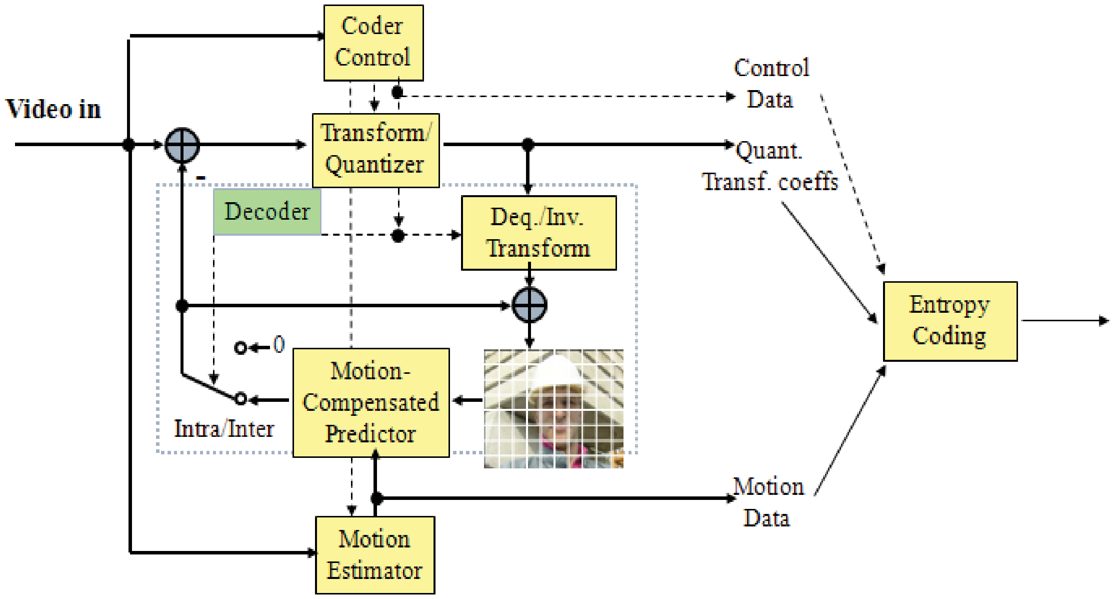


Figure 1.1 Block diagram of H.264/AVC with hybrid encoding.

The encoder includes two data parts, a forward path and a reconstruction path [56]. In the forward path, an input frame is coded in units of a 16x16 macroblock (MB). Each MB is coded in intra or inter mode. In either mode, a matching MB is generated from samples in the neighboring pixels that have previously encoded, decoded and reconstructed or from motion-compensated prediction from one or more reference frames. Then, the matching MB is subtracted from the current MB to produce a residual block, which is transformed and quantized to a set of coefficients. These coefficients are re-ordered and entropy encoded. The entropy encoded coefficients, together with side information required to decode the MB (such as MB mode, quantize step size, motion vector, etc) are passed to a Network Abstraction Layer (NAL) for transmission or storage. In the reconstructed path, quantized MB coefficients are decoded in order to reconstruct a frame for encoding of further macroblocks. The coefficients are re-scaled and inverse transformed to produce a difference block, which is not identical to the original residual block, for the quantization process introduces losses. The matching MB is added to the residual block to create a reconstructed macroblock (a distorted version of the original MB). Then, a filter is applied to reduce the effects of blocking distortion and reconstructed reference frame is created from a series of such reconstructed macroblocks.

H.264 defines a set of three Profiles, each supporting a particular set of coding functions targeting different applications. The Baseline Profile supports intra and inter-coding and entropy coding with context-adaptive variable-length codes (CAVLC). The Main Profile includes support for interlaced video, inter-coding using B-slice; inter coding using weighted prediction and entropy coding using context-based arithmetic coding (CABAC) [9]. Potential applications of the Baseline Profile include video telephone, video conferencing and wireless communications; potential applications of the Main Profile include television broadcasting and video storage.

1.1.2 Introduction to High Efficiency Video Coding (HEVC)

As video content is also growing in resolution and in recent a few years, HDTV and 4k2k content becomes the mainstream. Since 2010, a successor to H.264/AVC called High Efficiency Video Coding (HEVC) for next generation video coding is under standardization jointly by ISO/IEC MPEG and ITU-T VCEG [59]. HEVC aims to compress videos with twice compression efficiency compared to H.264/AVC High Profile at the encoding condition. The new standard is expected to be used in the future for super-high-vision broadcasting and video-delivery via limited bandwidth.

Different to previous standards, HEVC extend frame partition and hybrid prediction as well as residual encoding using a quadtree-based structure [11]. In frame partition, an input frame is divided into a sequence of largest coding unit (LCU) with a 64x64 block of luma samples together with the two corresponding blocks of chroma samples. The LCU concept is broadly analogous to that of the macroblock in H.264/AVC. The Coding Unit (CU) is the basic unit of region splitting used for inter/intra prediction. It is always square and it may take a size from 8x8 luma samples up to the size of LCU. The CU concept allows recursive splitting into four equally sized blocks, starting from the LCU. This process gives a content-adaptive coding tree structure comprised of CU blocks in variable sizes. The Prediction Unit (PU) is the basic unit used for carrying the information related to the prediction processes. In general, it is not restricted to being square in shape, in order to facilitate partitioning which matches the boundaries of real objects in the picture. Each CU may contain one or more PUs.

Due to the adoption of quadtree structure, available coding block sizes for hybrid prediction is twice of that in H.264/AVC and motion is compensated flexibly and accurately for different resolutions to achieve highest compression efficiency [12]. Accordingly, another quadtree formed by three levels of transform unit (TU) for DCT transform named residual quadtree transform (RQT) is also proposed in order to achieve the best tradeoff between energy compaction and adaptability [13].

In addition to the quadtree-based prediction and residual encoding, other advanced new coding tools such as directional intra prediction with 35 modes, residual quadtree transform, none-square motion partition, internal bit depth increase (IBDI), sample adaptive offset (SAO) [62][63], tiles and wavefront for parallelism [14] are also proposed for HEVC. By using these tools, by Feb. of 2012, HEVC achieves about 40% bit-rate saving compared to H.264 High Profile [15]. Figure 1.3 illustrates the

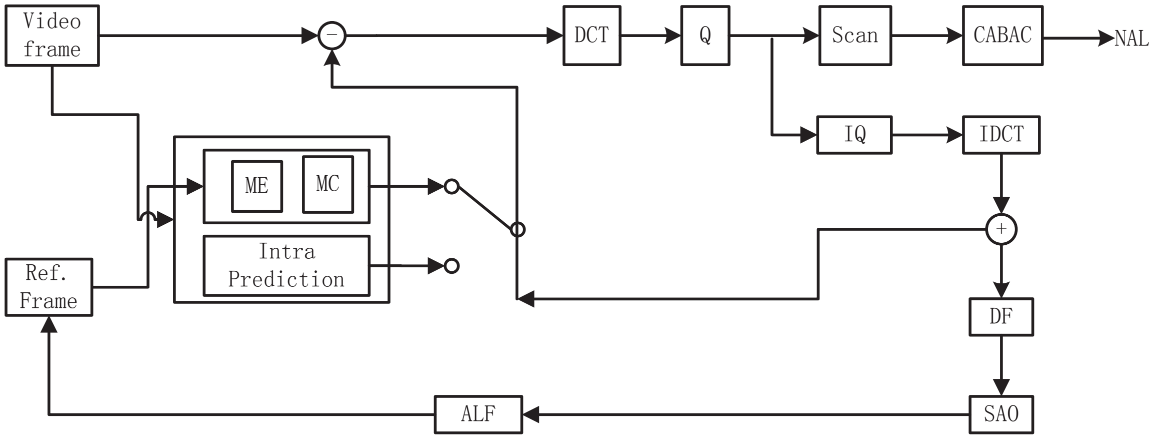


Figure 1.2 Encoder diagram of HEVC standard.

architecture of HEVC encoder.

Similar to H.264/AVC, HEVC encoder also includes a forward path and a reconstruction path. In the forward path, an input frame is coded in units of a 64x64 LCU. Each LCU is coded in intra or inter mode. In either mode, a matching LCU is generated from samples either in the neighboring pixels in current frame that have been reconstructed, or from motion-compensated prediction from other reference frames. Then, the matching LCU is subtracted from the current LCU to produce a residual block, which is transformed and quantized to a set of coefficients. These coefficients are re-ordered and entropy encoded. In the reconstructed path, quantized LCU coefficients are decoded in order to reconstruct a frame for encoding of further macroblocks. The remaining process of reconstructing a frame is identical to that in H.264/AVC, except that filtering using SAO and ALF are performed after de-blocking filters.

Target for different applications with the considerations of computational complexity, HEVC supports two coding scenarios: Low Complexity and High Efficiency shown in Table 1.1. The Low Complexity (LC) configuration is intended for low-delay applications and the High Efficiency (HE) configuration is intended to achieve high

Table 1.1 HEVC configurations of coding tools.

<table>
<tr><th>High Efficiency Configuration</th><th>Low Complexity Configuration</th></tr>
<tr><td colspan="2">Coding Unit tree structure (8x8 up to 64x64 luma samples)</td></tr>
<tr><td colspan="2">Prediction Units</td></tr>
<tr><td>Transform unit tree structure (3 level)</td><td>Transform unit tree structure (2 level)</td></tr>
<tr><td colspan="2">Transform block size of 4x4 to 32x32 samples (always square)</td></tr>
<tr><td colspan="2">Angular Intra Prediction (34 directions max.)</td></tr>
<tr><td colspan="2">DCT-based interpolation filter for luma samples (1/4-sample, 8-tap)</td></tr>
<tr><td colspan="2">DCT-based interpolation filter for luma samples (1/8-sample, 4-tap)</td></tr>
<tr><td colspan="2">Coding Unit based Skip & Prediction Unit based merging</td></tr>
<tr><td colspan="2">Advanced motion vector prediction</td></tr>
<tr><td>Context adaptive binary arithmetic entropy coding</td><td>Low complexity entropy coding phase 2</td></tr>
<tr><td>Internal bit-depth increase (2 bits)</td><td>X</td></tr>
<tr><td>X</td><td>Transform precision extension (4 bits)</td></tr>
<tr><td colspan="2">Deblocking filter</td></tr>
<tr><td>Adaptive loop filter</td><td>X</td></tr>
</table>

compression ratio for applications with enough hardware resources. HEVC encoder also considers about three kinds of temporal prediction structures depending on experimental conditions and reference picture list management depends on each temporal configuration [37]. The three structures are

1) Intra-only configuration. Each frame is encoded as IDR picture. No temporal reference pictures shall be used and it is not allowed to change QP within a picture.
2) Low-delay configuration. Only the first frame is encoded as IDR picture. The successive frames are encoded as P and B-picture (GPB). No picture reordering between decoder processing and output, with bit rate fluctuation characteristics.
3) Random-access configuration. Structure delay of processing units no larger than 8-picture group of pictures (GOP), dynamic hierarchical B usage with four levels Intra picture shall be inserted cyclically per about one second.

Combine the 2 coding configurations with the three temporal prediction structures, HEVC supports 6 kinds of coding configurations: Low Complexity Intra Only, High Efficiency Intra Only, Low Complexity Low Delay, High Efficiency Low Delay, Low

Complexity Random Access and High Efficiency Random Access. The computational complexity gradually increases for the 6 configures as their appearing order.

1.2 Effective but Computationally Intensive Coding Tools

In both standards, the most effective while computationally intensive tool is Rate Distortion Optimization (RDO) based hybrid encoding [60], which reduces spatial and temporal redundancy brutally trying a wide range of block types and prediction modes.

1.2.1 Mode Decision with Rate-Distortion Optimization in H.264/AVC

There are two indispensable and independent mode decision processes: Intra MD and Inter MD. Both MD processes are required in P and B frames.

Intra coding uses reconstructed but unfiltered neighboring MB to form predictor for intra blocks in I frame with the support from variable block sizes and multiple prediction modes. For luminance samples, the prediction block may be formed for each 4x4 block with 9 prediction modes or for an entire MB with 4 prediction modes.

Inter coding aims to reduce temporal redundancy in successive P and B frames. Variable block sizes ranging from 16x16 to 4x4 luminance samples are adopted for motion estimation and motion compensation. One MB can find its best matching block in several reference frames at quarter pixel resolution. Currently, inter coding supports 7 kinds of modes/block sizes ranging from 16x16 to 4x4.

The Rate Distortion Optimization (RDO) technique is applied to select the optimal block size and prediction mode to maximize coding quality and minimize data bits. The RDO process is shown in Figure 1.4. An RDO process utilized by H.264/AVC standard is like following. Let s and c denote source video signal and reconstructed signal from previous frames, respectively. The rate distortion cost between s and c, J, is

$$J(s,c,MODE\,|\,Q_p,\lambda_{MODE}) = SSD(s,c,MODE\,|\,Q_p) + \lambda_{MODE} * R(s,c,MODE\,|\,Q_p) \qquad (1.1)$$

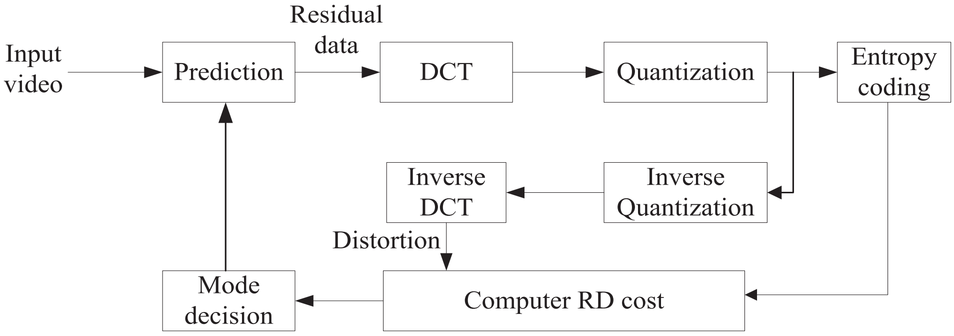


Figure 1.4 Rate Distortion Optimization process for Mode decision in H.264.

where λ_{MODE} is a Lagrange multiplier, given by $\lambda_{MODE} = 0.85 \times 2^{Q_p \div 3}$. *MODE* indicates any of modes supported. *R* represents number of bits from entropy coding process, including MB header, motion vector if any and all residual data associated with chosen *MODE* and QP. *SSD* reflects sum of squared differences between s and c

$$SSD(s,c,MODE\big|Q_p) = \Sigma_x \Sigma_y (s(x,y) - c(x,y,MODE\big|Q_p)) \qquad (1.2)$$

In order to compute RD cost for each available mode, operations of forward and inverse integer transform, quantization, and entropy coding is exhaustively performed. Thus, extremely high computation complexity is required by RDO process. Moreover, computation burden introduced by the rich set of block sizes/prediction modes is far more complicated than any other standards. According to our simulations, mode decision (MD) becomes the bottleneck in H.264 standard since it takes up at least 60% computation complexity of a whole encoder.

1.2.2 Quadtree-Structured Prediction and Residual Encoding in HEVC

The problem of mode decision becomes even critical in HEVC in that quadtree-structured coding block candidates for hybrid encoding have been increased by a factor of 2 and the prediction modes for intra coding has been increased to 35 modes. For an LCU, all the prediction units and available TUs for each quadtree level must be exhaustively searched by the introduced RDO process, this will cause an enormous increase in complexity compared to H.264.

The following experiments prove how critical the problem is. Experiments were performed using sequences with various motions, texture features in different resolution encoded at QP equal to 22, 27, 32 and 37. Table 1.2 lists the experimental results that

Table 1.2 Computational complexity of quadtree-based prediction and residual encoding.

Sequence	Sequences	QP	Complexity of Quadtree-based Prediction	Complexity of Residual Encoding
Class C (832x480)	Basketballpass	22	80.166%	19.834%
		27	81.768%	18.232%
		32	82.771%	17.229%
		37	83.415%	16.585%
	RaceHorses	22	75.903%	24.097%
		27	77.993%	22.007%
		32	79.313%	20.687%
		37	80.049%	19.951%
	Average	-	80.17%	19.83%
Class D (416x240)	BQMall	22	77.447%	22.553%
		27	79.383%	20.617%
		32	80.169%	19.831%
		37	80.473%	19.527%
	PartyScene	22	74.144%	25.856%
		27	76.962%	23.038%
		32	78.432%	21.568%
		37	79.911%	20.089%
	Average	-	78.37%	21.63%
Class E (1280x720)	Vidyo1	22	73.141%	26.859%
		27	75.930%	24.070%
		32	76.436%	23.564%
		37	77.714%	22.286%
	Vidyo3	22	73.341%	26.659%
		27	74.930%	25.070%
		32	76.622%	23.378%
		37	77.813%	22.187%
	Average	-	75.74%	24.26%

75-80% complexity was taken by the prediction process, for all tested sequence at all QP values. Thus, fast algorithms to optimize the HEVC encoder are urgently required.

1.3 Performance Evaluation Methodology for Video Compression

There are three important parameters widely used for encoder performance evaluation in terms of video visual quality, encoded bits need to be transmitted or storage, and encoding complexity required to achieve the former two.

The peak-signal-to-noise-rate (PSNR) is mostly commonly used as a measure of visual quality of reconstruction of lossy compression codec. The signal in this case is the original data and the noise is the error introduced by compression. PSNR is calculated via mean squared error between original signal and reconstruction (MSE) as

$$PSNR = 10 \cdot \log_{10}(\frac{255^2}{\overline{MSE}}) \tag{1.3}$$

Given a noise free image *I* and its noisy reconstruction *R*, *MSE* is given by

$$MSE = \frac{1}{mn} \cdot \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - R(i,j)]^2 \tag{1.4}$$

In compression of color video consisted of three channels, $\overline{MSE}$ is calculated using *MSE* in luminance Y and chrominance U and V channels together. Thus, we have

$$\overline{MSE} = \frac{4 \cdot MSE_Y + MSE_U + MSE_V}{6} \tag{1.5}$$

Typical values for the PSNR in lossy image and video compression are between 30 and 50 dB [16], where higher is better. Acceptable values for wireless transmission quality loss are considered to be about 20 dB to 25dB [17].

The encoded bits in telecommunications and computing are usually called bit rate, which is the number of bits that are conveyed or processed per unit of time and it is quantified using the bits per second (bit/s) unit. The encoding complexity can be measured by the software encoding time of running a video encoder.

The difference of PSNR, BR and coding time denoted as $\Delta PSNR$, ΔBR and $\Delta Time$ respectively between proposed algorithm and the full brutal search strategy are tested under the conditions and the same sequences. They are defined as

$$\begin{aligned} \Delta PSNR &= PSNR_{pro} - PSNR_{ori} \\ \Delta BR &= BR_{pro} - BR_{ori} \end{aligned} \tag{1.6}$$

where $PSNR_{pro}$ and BR_{pro} is the evaluation of proposed algorithm and $PSNR_{ori}$ and BR_{ori} is the evaluation from original video encoder. $\Delta PSNR$ and ΔBR can also be calculated as BD-PSRN and BD-BR using the numerical averages between the RD curves using four QP values [18]. $\Delta Time$ is calculated by

$$\Delta Time = \frac{EncTime_{pro} - EncTime_{ori}}{EncTime_{ori}} \cdot 100\% \tag{1.7}$$

where $Time_{pro}$ and $Time_{ori}$ are the simulation times of running an optimized and original video encoder, respectively.

1.4 Main Contributions

In this dissertation, algorithms and optimization frameworks are proposed to reduce the heavy computational burden in RDO-based hybrid prediction in both the H.264/AVC and HEVC standards. Our main contributions are listed as follows.

1) An accurate and fast mode decision scheme is proposed for H.264/AVC which is proved to be capable of reducing 52.5% computational complexity of H.264/AVC encoder without sacrificing video quality and encoded bits.

2) A content adaptive hierarchical decision approach of variable coding blocks is proposed for compression 4k2k videos for HEVC encoder. Acceleration can be achieved for all kinds of videos which contain largely different characteristics in texture and motion.

3) A novel AZB detection algorithm is proposed with high detection accuracy for different block sizes. Based on AZB detection, an optimization scheme is brought forward to reduce computational complexity and hence power consumption for HEVC encoder. The proposed detection method consumes ultra low complexity and experiments show that the scheme reduces the computational complexity of HEVC by up to 70.46% and 50.34% on average.

1.4.1 Accurate Mode Decision for H.264/AVC Hybrid Prediction

This proposal brings forward a novel and accurate MB homogeneity detection method to accelerate H.264/AVC intra and inter coding. Based on homogeneity detection, 16x16 or 4x4 block size is appropriately selected for intra coding; meanwhile, either the large blocks in {16x16, 16x8, 8x16} or sub-blocks in {8x8, 8x4, 4x8, 4x4} are chosen for inter coding. Simulations demonstrate that consistent encoding gain is achieved for all videos with different motion and spatial features. Complexity for intra coding alone can be reduced by 31%-34% and time saving for overall mode decision is 52.5% with PSNR drop always less than 0.06dB and bit rate increment less than 0.9%.

In intra coding, compared with [19] and [20], a further of 19% and 8% complexity reduction is achieved by proposed method with the same performance in PSNR and BR. Especially, for sequences with very complex background, proposed algorithm performs much better than [19] and [20].

In inter coding, compared with a milestone [21] (IEEE Trans. of CSVT 2005), our proposal always achieves consistent gain from 39% to 46% in complexity reduction for all videos with different characteristics; while [21] only accelerates 10% for sequences with fast motions or complicated background. As a total scheme for mode decision of hybrid prediction, compared with [22] (IEEE Trans. of CSVT 2009), our mode decision algorithm reduces an extra 11.95% coding time, meanwhile, it achieves better performance in BR by saving an extra of 0.55% bitrate at the same PSNR.

The reason for our proposal superior to previous MB homogeneity-based methods is that proposed method extracts entropy feature directly from all the raw pixels which explores the spatial nature of MB. Thus, consistent coding gain is achieved for various kinds of videos with different spatial and motion characters.

1.4.2 Content Adaptive Decision of Variable Block Sizes for HEVC for High Resolution Videos

A content adaptive decision approach is proposed to decide variable coding blocks for 4k2k videos using HEVC encoder. This proposal explores features in temporal, spatial and transform domains for acceleration. Before encoding starts, analysis on utilization ratio of each coding depth is performed to skip rarely adopted coding depths at frame level. Texture complexity (TC) measurement is applied to filter out none-contributable coding blocks for each largest coding unit (LCU). In this step, a dynamic threshold setting approach is proposed to make filtering adaptable to videos and coding parameters. During encoding process, sum of absolute quantized residual coefficient (SAQC) is used to prune useless coding blocks for CUs. By using proposed scheme, motion estimation is performed for prediction blocks within a narrowed range.

Experiments show that proposed scheme speeds up original HEVC by a factor of up to 61.89% and by an average of 33.65% for 4kx2k video sequences with PSNR degradation less than 0.01dB and bit increment less than 0.4%.

Compared with a proposal [23] (JCTVC-F092 2011) adopted by HEVC, our approach reduces 15.8% more complexity with slightly increased BR. The reason is that for sequences with complex texture, motion and for low QP cases, complexity reduction from [23] is deteriorated because [23] only takes advantage of SKIP mode while SKIP mode happens infrequently for complex texture, motion and for low QP. On the other hand, our proposal utilizes multiple features so that it can reduce a large proportion of complexity for sequences with a wide range of content characteristics.

1.4.3 Ultra Low Complexity AZB-Based Optimization for Quadtree-Structured Prediction and Residual Encoding in HEVC

This part proposes the utilization of AZBs to constrain exhaustive partitions in quadtree-based prediction and residual encoding in HEVC standard.

This optimization scheme consists of three parts. First, a novel condition is derived to detect variable-sized AZBs in all Y and UV channels. The, using the condition, an early termination technique is presented to reduce search points of motion estimation (ME) for a wide range of PUs in size of {64×32, 32×64, 32×32, 32×16, 16×32,16×16, 16×8, 8×16, 8×8 8×4, 4×8}. More importantly, AZBs are fully used to prune non-contributing PUs in a two-dimensional manner. The pruning can be beneficial for both homogenous areas and regions with complex or fast motions. As a result, up to 70.46% of coding complexity (50.34% on average) is reduced with negligible performance loss. The proposed scheme is appropriate for implementation in hardware to save power.

Compared with Sousa's method [24], in our proposed AZB detection algorithm, detection accuracy for luminance blocks is increased by 48.51%, 28.26%, and 8.43% for 8×8, 16×16, and 32×32 AZBs, respectively; meanwhile, for chrominance blocks, the increases of detection accuracy for 4×4, 8×8, and 16×16 blocks are 13.43%, 38.14%, and 42.84%, respectively. In addition, proposed method consumes ultra low complexity which makes it appropriate for implementation in hardware.

Compared with Shen's work [25], the newest work for HEVC encoder optimization, our proposal achieves better coding performance than [25] in terms of BD-BR and BD-PSNR: BD-BR loss in our proposal is improved by 1.14% less and meanwhile, the average coding complexity reduction of our proposal is approximately 3% higher. The reason is that [25] trains features using some video samples and as the training process cannot cover all possible cases, its performance will be affected when features of input videos are different from those of training samples.

1.5 Dissertation Organization

The proposed three optimization schemes have the same target which is to select only contributing proper block sizes for RDO so as to reduce overall encoding complexity in H.264/AVC and HEVC standards.

The correlation among the three proposals can be summarized in Table 1.3. The first proposal can be directly applied to HEVC without modification. However, since it mainly chooses the best coding block sizes from {16x16, 4x4} for intra prediction , and {16x16, 16x8, 8x16, 8x8,8x4,4x8,4x4} for inter prediction, it can only accelerate encoding process in depth 2 and 3 in HEVC.

The second and the third proposals can reduce the encoding complexity of HM 4.0

Table 1.3 The correlation among the three proposals

Proposals	Target standard	Features utilized	Process been accelerated
MB homogeneity detection based mode decision scheme	H.264/AVC (JM 12.4)	Texture homogeneity in spatial domain	Intra and inter prediction
Content adaptive hierarchical decision of variable blocks	HEVC (HM 4.0)	Texture homogeneity in spatial domain; Utilization ratio of each depth in temporal domain; SAQC in transform domain	Intra and inter prediction
AZB-based optimization scheme for quadtree-based prediction and residual encoding	HEVC (HM 6.0)	AZB detection in spatial domain; SAQC in transform domain	Intra and inter prediction; residual encoding

and HM 6.0 by 33.4% and 50.3%, respectively. It is possible to integrate them together but modifications of how to integrate are needed. And the complexity reduction by integrated scheme should be larger than 50.3% but less than the sum of 33.4% and 50.3%.

The rest of this dissertation is organized as follows. Chapter 2 investigates previous works on mode decision and presents a new fast mode decision scheme to reduce computational complexity in hybrid prediction for H.264/AVC standard based on texture analysis. Chapter 3 brings forward an adaptive scheme that exploits feature in temporal, spatial and transform domains to speed up the original quadtree-based prediction, targeting at videos in resolution of HD and beyond. Chapter 4 presents a solution which takes full advantage of AZB for acceleration for HEVC encoder, a novel and ultra low complexity AZB detection method is also included in this chapter. Chapter 5 concludes the contribution of this dissertation.

2. MODE DECISION SCHEME FOR H.264/AVC

2.1 Background on Mode Decision

This section introduces the basic knowledge of prediction block sizes for inter and intra prediction, and allowable prediction modes for intra prediction.

2.1.1 Block Types and Prediction Modes in Intra Prediction

Intra coding is conducted in spatial domain instead of in transform domain for the first time, because referencing MB in raster scan is much more efficient than in frequency domain. Moreover, it uses reconstructed but unfiltered neighboring MBs to form predictor, to get better compression for intra blocks in P-slice. Most importantly, variable block sizes and multiple prediction modes are supported.

For luminance samples in intra coding in baseline and main profiles, the prediction block may be formed for each 4x4 block (denoted as I4MB) or for an entire MB (denoted as I16MB). The 4x4 prediction is suitable for images with significant details. It supports 8 directional modes plus one DC mode. The I16MB, which is well suited for smooth image areas, supports 4 prediction modes. For the chrominance 8x8 block of a macroblock (denoted as C8MB), it is always predicted using a similar prediction technique as for I16MB.

For 4x4 intra coding, one 4x4 block can be predicted across 9 directions using the neighboring pixels from *A* to *Q* if available, shown in Figure 2.1(a). The pixels from *A* to *Q* have previously been encoded and reconstructed. The predicted samples for current pixels *a*-*p* are formed from a weighted average of prediction samples *A* to *Q*. The nine prediction modes for 4x4 intra coding, indicated by arrows in Figure 2.1(b) are:

Mode 0: vertical prediction

Mode 1: Horizontal prediction

Mode 2: DC prediction

Mode 3: Diagonal down-left prediction

Mode 4: Diagonal down-right prediction

Mode 5: Vertical-right prediction

Mode 6: Horizontal-down prediction

Mode 7: Vertical-left prediction

Mode 8: Horizontal-up prediction

Note that DC prediction (mode 0) is modified depending on which samples *A*-*M* are available. The other modes (1-8) may only be used if all of the required prediction samples are available.

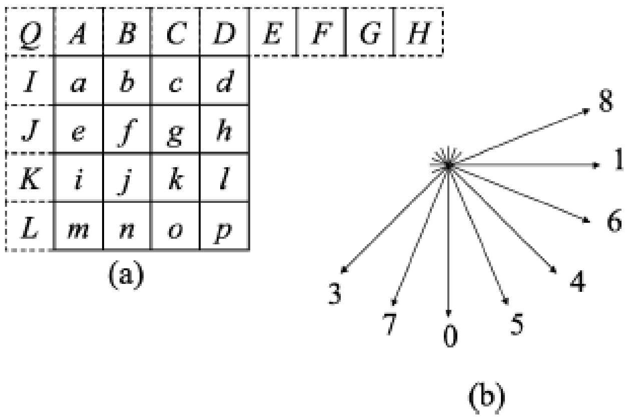


Figure 2.1 (a) A 4x4 block and its neighboring pixels (b) Directions of the 8 prediction modes

For 16x16 intra coding, one MB is coded as an entire block. Generally speaking, intra 16x16 prediction is suitable for smooth MBs where a uniform prediction is conducted for the whole luminance component of an MB. Four prediction modes are supported as below. Plane mode is based on a linear spatial interpolation. It is non-directional mode but has some directional characteristics.

Mode 0: Vertical prediction

Mode 1: Horizontal prediction

Mode 2: DC prediction

Mode 3: Plane prediction

In commonly used 4:2:0 YUV videos, each MB is associated with two 8x8 chrominance components. In intra coding in H.264, both these two components share the same prediction mode. The available prediction modes for C8MB are the same as I16MB but in different order.

In summary, if an MB is encoded in intra mode, the following procedure is performed. All the 9 modes for 4x4 block and all the 4 modes for 16x16 and chrominance blocks are exhaustively examined to get the minimum RD cost. Each MB needs 4*(16*9+4) =592 RDO calculation. If the target application is HDTV, each frame needs 1920*1088*592/256=4,830,720 RD calculations. This is very hard to be implemented in real-time. Thus intra coding strongly needs to be fastened.

2.1.2 Variable Block Sized Prediction in Inter Prediction

Inter coding aims to reduce temporal redundancy in successive P and B frames in natural videos. In H.264, variable block sizes ranging from 16x16 to 4x4 luminance samples are adopted for motion estimation and motion compensation. One MB can find its best matching block(s) in previous several reference frames at quarter pixel resolution.

Currently, inter coding supports 7 kinds of modes/block sizes ranging from 16x16 to 4x4(16x16, 16x8, 8x16, 8x8, 8x4, 4x8 and 4x4). One reason for adopting 7 different block sizes in H.264 inter coding is to represent scene movement more accurately so as to reduce temporal redundancy efficiently. These 7 modes forms a 2 level tree structure. The level 1 consists of blocks {16x16, 16x8, 8x16} and level 2 consists of blocks {8x8, 8x4, 4x8, 4x4}. Figure 2.2(a) illustrates a whole frame partition and MB partition style for QCIF picture. It can be seen that one MB can be partitioned and compensated into combinations of different block sizes.

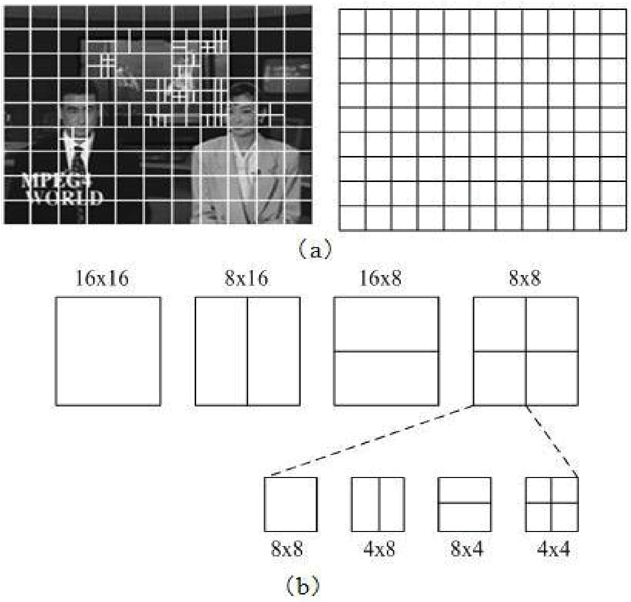


Figure 2.2 (a) Frame partition example for QCIF video in H.264/AVC.
(b) Macroblock and sub-block partitions in H.264/AVC inter coding.

A separate motion vector is required for each partition. Each motion vector must be coded and transmitted. Choosing level-1 partition means that a small number of bits are required to signal motion vectors and the type of partition. However, the motion compensated residual may contain a significant amount of energy in frame areas with high detail. Choosing level-2 partition may give a lower-energy residual after motion compensation but requires a larger number of bits to signal the motion vectors and partitions. Hence, the choice of partition size has a significant impact on compression performance. In general, a large partition size is appropriate for homogeneous areas of the frame and a small partition size may be beneficial for detailed areas.

2.1.3 Previous Mode Decision Algorithms

The "trying all and select the best" mode decision philosophy in intra and inter coding is able to find the optimal block size and prediction mode. However, by doing simulations, it is found that homogenous MBs almost always are coded by large block size after RDO selection. Sub-blocks have high potential to code MBs with detailed information [19]-[21] [26]-[28]. Also, there are lots of homogenous areas/regions in natural videos, up to 35%. Based on this observation, many methods are proposed [19]-[21] [26]-[28] to reduce coding complexity and to save coding time.

2.1.3.1 Fast Intra Acceleration Algorithms

It is believed that fast intra prediction algorithms are very important in reducing the overall complexity of H.264/AVC. There are two types of fast algorithms: block size decision and mode decision algorithms.

Intra block size decision algorithms aim to predict the optimal coding block instead of trying all in force manner. The concept behind block size decision is that the optimal block size highly correlates to the homogeneity of MB [19] [20]. If one MB is smooth it has very high potential to be encoded in 16x16 block size; otherwise if it contains complex or detailed information it is better to be encoded in 4x4 block size. Thus, the key point for block type decision algorithm is to extract the smoothness of MBs.

Zhang [19] uses amplitude of filtered edge vector associated with SATD for MB homogeneity measuring. Problem with this method is that it only supports CIF videos and can only gain about 10%-18% computation reduction. Work [20] uses simplified DCT transform to get the AC/DC ratio. Next, the logarithm value of the AC/DC ratio shown in Eq.(2.1) is defined as the smoothness of this MB for the purpose of block size decision.

$$AC\_DC\_Ratio = \frac{\log[(\Sigma_{m=0}^{7}\Sigma_{n=0}^{7}a_{mn})^2 - \frac{1}{8\times 8}(\Sigma_{m=0}^{7}\Sigma_{n=0}^{7}a_{mn})^2]}{\log[(\frac{1}{8\times 8}(\Sigma_{m=0}^{7}\Sigma_{n=0}^{7}a_{mn})^2)\cdot 64]} \qquad (2.1)$$

This method gains significant complexity reduction. However, bit-rate increase is still high compared with similar work.

Intra mode decision algorithms aim to predict promising modes out of full-search. Basically, two different kinds of fast intra mode decision algorithms exist.

The first category explores edge directions contained in one MB in pixel domain. Obviously, various filters in image processing field can achieve this purpose. Pan's algorithm [29] is the most representative work and also a milestone. In his work, edge detection histogram for each mode firstly is calculated. Then the maximum one and its two neighbors plus DC mode are selected as candidate modes. Totally, this method can reduce 63% coding complexity for all-I sequences. However, histogram calculation is quite time consuming itself. In addition, wrong decision may happen due to the accumulation of edge strength in each pixel, which cannot indicate edge directions sometimes. Intra modes only need the edge direction but not the detailed edge detection in every pixel. This is the reason why the RD performance tends to be damaged. Similar work and improvements based on Pan’s algorithm can be found in [30]-[33]. These algorithms also bases on edge direction extraction and they still needs 4 candidate modes for 4x4 blocks. Consequently, processing cycles in hardware and computational complexity in software are still high. The advantage of pixel-domain based edge extraction methods is that computation time can be saved significantly. However, they unavoidably suffer destroyed coding performance in PSNR and BR [30]-[33].

Very few papers belong to the second category which explores edge direction in transform domain. Usually, DCT transform is a good choice. In [34], DCT transform for 4x4 and 16x16 blocks are performed respectively to get the directional information in horizontal, vertical and diagonal information. Then, a mapping from edge direction to intra modes will be done to choose suitable modes for final intra coding. Work [34] is able to achieve better coding performance than pixel-domain methods while computation reduction is much less.

2.1.3.2 Inter Block Size Decision Algorithms

Inter coding supports 7 modes including SKIP mode. Exhaustive searching manner is obviously a waste of time. Previous work on inter mode selection can be classified into 2 categories.

The first category utilizes temporal correlation between co-located MB and current MB to select promising block sizes. The basic assumption is that current MB has high potential to re-use the best mode which codes the co-located MB. Work [35] is one representative work. In this work, co-located MB’s best mode is set as initial mode for current MB. This initial mode becomes the best mode when its RD cost for current MB is smaller than that from co-located MB. Otherwise, other smaller blocks will be added as candidates one by one. Similar works which explore temporal correlation can be found in [21] and [36]-[37], [58]. Work [21] explores stationary correlation for co-located MBs and stationary refers to “stillness” between consecutive frames in temporal dimension. It believes that image background exhibits lots of stationary areas and these regions tend to be coded using big block size after RDO computations. As to measure stationary feature, it uses sum of absolute differences (SAD) to check whether one MB changes or not. The SAD is defined as

$$SAD = \sum_{i=1}^{16}\sum_{j=1}^{16} abs(M(i,j) - N(i,j)) \tag{2.2}$$

Here, $M(i,j)$ and $N(i,j)$ represent pixel intensities in the previous MB and current MB, respectively. If SAD is less than a threshold, the MB is classified as stationary. This method can save some limited coding time. Work [36] focuses on selective intra coding, which is to select intra coding only when it have to and all inter modes have to be performed by RDO process. Temporal correlation is defined as the amount of texture bits for the best inter mode; spatial correlation is calculated as the sum of absolute difference (SATD) between pixels at a boundary of current and its adjacent upper and left encoded blocks. This work at most can save 11.9% coding time, since the probability of MBs coded in intra mode in P/B slices no more than 20% theoretically. The disadvantage of above methods is that temporal correlations become unapparent in fast or complex motion videos, such as “football” and "mobile". In other words, coding performances vary sharply among videos with different level of motions [21].

The second category measures MB's complexity to select proper block sizes. Previous research [26] [27] use AC energy variance of DCT transform to measure MB's homogeneity for block size selection. Sum of amplitude of edge vectors is also used in [21] for inter block size selection. In work [26], homogeneity on entire 16x16 block and its 2 8x8 blocks are measured respectively, to select either level 1:{16x16,16x8,8x16} or level 2: {8x8, 8x4, 4x8, 4x4}. That is homogeneity detections are needed twice. As a result, at most 30% reduction can be achieved.

2.2 Fast Intra Mode Decision Algorithm for H.264/AVC

2.2.1 Observation

The multiple prediction modes are indicated by arrows with different directions shown in Figure 2.1(b). Predictor for each available block size is formed from a weighted average of the neighboring samples *A-M*. Optimal prediction mode is the one which minimizes the residual between predictor and source block. Except DC mode, the other 8 directional modes indicate luminance variation in certain direction, which becomes edge direction. This is the fundamental observation for intra mode decision algorithms. Edge extraction information is helpful to eliminate futureless modes for intra coding.

2.2.2 Edge Detection based Fast Mode Decision

Intra coding should be optimized for two reasons (1)The cost of computing RD cost of intra mode is about 5 times higher than for inter modes [36]; (2)Intra modes are also exhaustively examined in P and B slice. The proposed algorithm uses Prewitt operators to extract the two most dominant edge directions of each block; the two directions are utilized to reduce 9 modes to only 3 modes. Proposed mode decision method can be divided into 3 steps: block down-sampling, local edge extraction and mode generation

Step 1: Down sample input luminance 4x4 block into 3x3 size. Each of the target coefficients of the down sampled 3x3 block, denoted as a_i (i = 1, 2 ... 9), is the average value of 4,16, 64 original pixels in I4Mb, C8MB and I16MB, respectively. This step needs only addition and shift operations.

Step 2: 4 Prewitt operators specified in Figure 2.3, are used to extract the local edges in vertical, horizontal, diagonal 45^o and diagonal 135^o directions, respectively. Correspondingly, these four directions are mode 0, mode 1, mode 3 and mode 4. For each direction, calculate the edge response of the 3x3 block to its corresponding operator by $response_{v,h,45,135} = \sum_{i=1}^{9} a_i \cdot operater_{v,h,45,135}[i]$ .Here, *v, h, 45* and *135* denote vertical, horizontal, diagonal 45° and 135°, respectively. Next, pick up the maximum and second maximum value from the above 4 responses, denoted as *Max* and *SecondMax. T*his step needs addition and comparison operations.

Horizontal

-1	-1	-1
0	0	0
1	1	1

Vertical

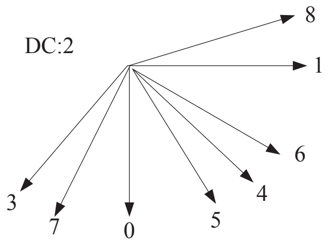
-1	0	1
-1	0	1
-1	0	1

45 degree

-1	-1	0
-1	0	1
0	1	1

135 degree

0	1	1
-1	0	1
-1	-1	0



DE is 0; SDE is 3 select 0, 7 and DC modes
DE is 0; SDE is 1 select 0, 5 and DC modes
DE is 1; SDE is 3 select 1, 8 and DC modes
DE is 1; SDE is 4 select 1, 6 and DC modes

Figure 2.3 Prewitt Operators and selection rule for Intra mode decision algorithm

It is believed that the mode corresponding to the maximum edge response is a dominant edge and a good predictor for intra prediction [29] [32]. However, if the two modes adjacent to the dominant edge were mode candidates [29] [32], it is obvious that the mode which locates between the most dominant edge (DE) and second most dominant edge (SDE) has higher possibility to be a good predictor than the other mode .

Step 3: This step decides which modes will be selected.

(1) The mode corresponding to DE will be chosen as candidate 1;

(2) The mode closely adjacent to candidate 1, locating between candidate 1 and the mode corresponding to *SDE,* will be chosen as candidate 2;

(3) DC mode is always chosen as candidate 3.

For example, if mode 0 corresponds to DE and mode 3 corresponds to SDE, mode 0, 7 and DC mode will be chosen. The bottom part of Figure 2.3 shows the selection rule. Using this rule, only 3 modes are selected. Thus, proposed fast mode decision algorithm can further speed up intra prediction.

The mode decision for luminance 16x16 and Chrominance 8x8 blocks are similar to that for I4MB. The only difference is that 135^{o} is no longer need. Consequently, only 2 modes are needed for these 2 block sizes: the mode corresponding to *DE* and DC mode.

2.2.3 Experimental Results

Our proposed algorithm is tested within JM 12.4, on a PC with 2.4G Hz CPU and 1 GB memory. Rate distortion optimization is enabled and QP is set to 28. Three measurement parameters, △PSNR, △BR and △time are used to evaluate the coding performance.

Table 2.1 and 2.2 show the comparison result with [29] for all-I frame QCIF and

Table 2.1 Coding performance of proposed mode decision algorithm on QCIF

Sequence	Pan's algorithm[29]			Proposed algorithm		
	△PSNR(dB)	△BR(%)	△Time(%)	△PSNR(dB)	△BR(%)	△Time(%)
Foreman	-0.285	4.437	-65.378	-0.098	4.616	-72.914
News	-0.294	3.902	-55.339	-0.094	3.933	-73.858
Container	-0.234	3.695	-56.357	-0.089	3.670.	-73.692
Silent	-0.183	3.540	-65.170	-0.076	3.858	-72.931
Coastguard	-0.106	2.361	-55.026	-0.095	2.437	-73.880
Average	-0.220	3.587	-59.454	-0.090	3.702	-73.455

Table 2.2 Coding performance of proposed mode decision on CIF

Sequence	Pan's algorithm[29]			Proposed algorithm		
	△PSNR(dB)	△BR(%)	△Time(%)	△PSNR(dB)	△BR(%)	△Time(%)
Paris	-0.230	3.21	-57.779	-0.074	2.661	-77.134
Mobile	-0.255	3.168	-59.086	-0.091	2.893	-76.514
Tempete	-0.299	3.514	-57.697	-0.080	2.504	-76.564
Bus	-0.218	3.849	-58.118	-0.086	3. 174	-77.029
Stefan	-0.242	3.717	-57.972	-0.102	2.351	-75.538
Average	-0.236	3.492	-58.130	-0.087	2.712	-76.556

CIF sequences, respectively. The tested frame is 300 and frames are I frames. In QCIF case, proposed algorithm can save an average of 73.455% encoding time for intra prediction, with average increment of bitrate 3.7% and loss of PSNR 0.090. Compared with [29], our proposal can achieve better performance in PSRN. In CIF case, proposed algorithm is able to save about 76.556% encoding time, with average increment of bitrate 2.7% and loss of PSNR is 0.087. Compared with [29], PSNR is with much better performance. As a summary, proposed algorithm achieves a further time reduction of 14.001% and 18.426% for QCIF and CIF sequences.

2.3 Block Type Decision Algorithm for Intra Prediction in H.264/AVC

2.3.1 Obervations

It is observed that I4MB is suitable for images with significant details and I16MB prediction is well suited for smooth areas. Also it is observed that in natural video sequences, there are a lot of homogeneous regions. It is clear that complex MBs in Figure 2.4 (b) are coded by I4MB, e.g. audience sitting in stand and the player's head

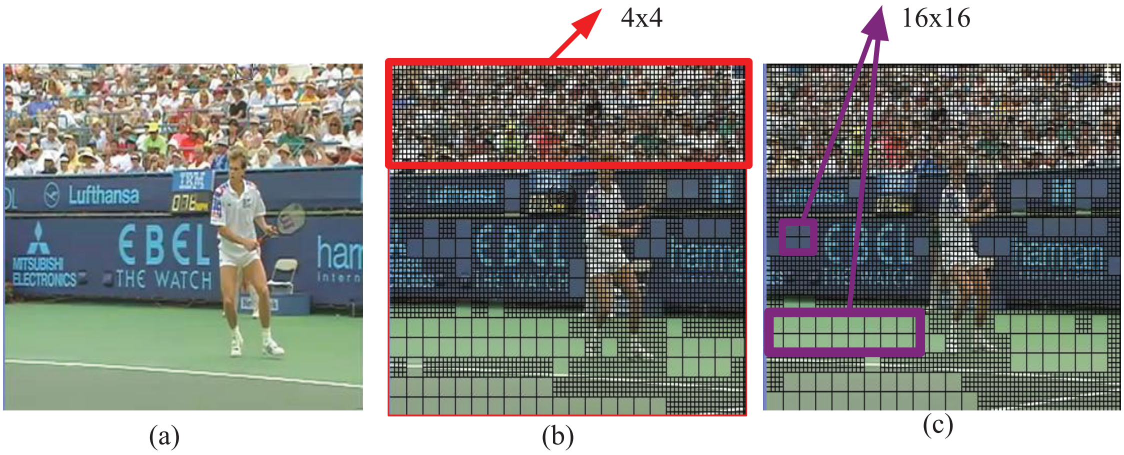


Figure 2.4 (a) "Stefan" original I frame. (b) MBs marked in red are coded by I4MB. (c) MBs marked in purple are coded by I16MB

and body. On the other hand, homogeneous shown in Figure 2.4(c) areas are coded by I16MB, e.g. part of racket court and billboard. So it is without doubt that if we can judge MB's homogeneity accurately we can safely skip unnecessary block sizes so as to reduce on necessary encoding complexity.

2.3.2 Homogeneity Detection and Block Type Decision

Shannon entropy is famous in measuring of information for its average minimum number of bits needed for storage or communication, based on the frequency of each symbols. It is defined as

$$H(X) = -\sum_{x \in X} p(x) \log p(x) \tag{2.3}$$

where *p(x)* indicates the frequency of symbol *x*. An important property of entropy is that it is minimized only when all the messages in the message space are equal. Intuitively, we take the intensities of all the 256 pixels of one MB to represent set *X* and extract the entropy feature of this MB as

$$H = -\sum_{i=0}^{255} p(Y(i)) \log p(Y(i)) \tag{2.4}$$

$Y(i)$ denotes the intensity of pixel i. $p(Y(i))$ denotes the probability that $Y(i)$ appears. According to the above property, if all the pixels in one MB have the same intensity, this MB's entropy feature will be 0. On the other hand, if these 256 luminance value are totally different, above entropy feature equals to 8. So the upper and lower limit for a MB's entropy feature is

$$0 \le H \le 8 \tag{2.5}$$

Figure 2.5 (a) and (b) shows entropy distribution of the first 5 I frames from “foreman” with 1980 entropy values and QP are 20 and 40, respectively. It is observed that I4MBs shown in red in most cases have apparently higher entropy values than that of I16MBs, although a few entropy values from both block sizes are overlapped. This observation functions as a solid base for proposed block size decision method.

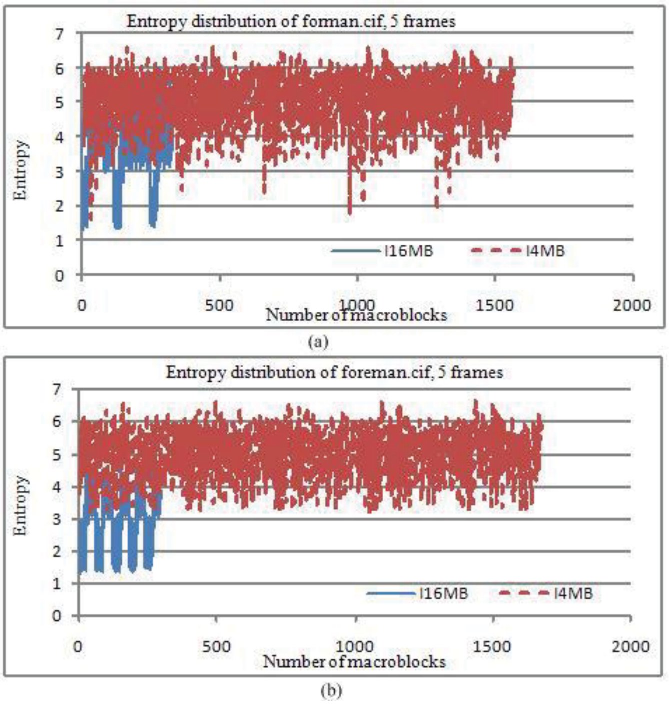


Figure 2.5 (a) Entropy distortion of I16MB and I4MB of forman.cif, QP=28
(b) Entropy distribution of I16MB and I4MB of foreman.cif, QP=40

Figure 2.6(a) shows the full search encoding flow for original intra coding. According to the above observations, we can use two thresholds, if denoted as α and β, to distinguish I4MBs and I16MBs. The decision rule is: if entropy of one macroblock *H* is larger than β, than only 4x4 block type is used for intra coding; if *H* is smaller than α, only 16x16 block type is used for intra coding; otherwise both block types are used to

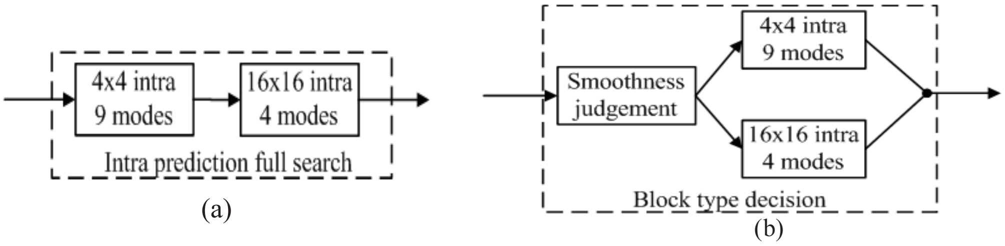


Figure 2.6 (a)Full search of intra prediction (b)Intra block size decision flow

ensure video quality. Figure 2.6(b) shows the optimized encoding flow with help from block type decision.

One issue is how to set the thresholds. Since the entropy is pre-extracted before intra coding, it will not change with any coding parameters. Thus, it is reasonable to use fixed thresholds to do block type decision. Distribution in Figure 2.5 (a) and (b) are similar, since entropy is pre-extracted directly from image pixels. Thus, we can use fixed α and β to distinguish 16x16 and 4x4 blocks.

2.3.3 Experimental Results

When α is set to 3.2 and β to 4.6, we get the decision accuracy and encoding performance of proposed algorithm for various CIF sequences, shown in Table 2.3. The block type decision accuracy equals to the ratio of correctly decided MBs to the total MBs. It is disclosed that decision accuracy of proposed method is always higher than 90% and averagely it is 94.88%.

Table 2.3 Block type decision accuracy based on entropy

Sequence	Number of I16MB		Number of I4MB		Accuracy
	JM	Proposal	JM	Proposal	
Foreman	411	388	1569	1592	94.65%
News	620	646	1360	1334	96.06%
Mobile	68	76	1912	1904	98.33%
Akiyo	880	771	1100	1209	90.20%
Container	796	775	1184	1205	92.98%
Coast	65	27	1915	1953	97.58%
Mother	748	772	1232	1208	94.39%
Average	512	494	1478	1486	94.88%

Table 2.4 shows encoding performance using proposed block type decision algorithm. Seven CIF video sequences which contain different features in texture and motion are tested. The left hand columns show encoding performance comparison with JM 12.4. It is shown that the average ΔPSNR is about -0.001 dB with bit-rate increased about 0.648%. At the same time, average reduction of encoding complexity is 33.4%. The right

Table 2.4 Encoding performance of proposed block type decision algorithm for CIF

Sequence	Compared with JM			Compared with [19]		
	△PSNR(dB)	△BR(%)	△Time(%)	△PSNR(dB)	△BR(%)	△Time(%)
Foreman	+0.001	0.36%	-31.672%	+0.002	+0.17%	-20.472%
News	+0.001	0.55%	-34.073%	+0.002	+0.09%	-16.403%
Mobile	+0.001	0.04%	-34.174%	+0.002	-0.01%	-20.984%
Akiyo	-0.012	1.35%	-37.879%	-0.001	+1.27%	-21.279%
Container	+0.003	0.59%	-31.062%	+0.020	-0.30%	-20.702%
Coast	+0.002	0.10%	-31.116%	+0.003	+0.03%	-20.926%
Mother	-0.002	1.51%	-33.857%	+0.003	+1.36%	-15.567%
Average	-0.001	0.648%	-33.4%	+0.003	+0.37%	-19.48%

hand columns demonstrate comparison with [19], in which sum of sobel-filtered amplitude is used for smoothness judgment. Compared with [19], for any of tested sequences, encoding complexity reduction is from 15.56% to 21.27% and average reduction is 19.48%. At the same time, better PSNR of 0.003 dB is achieved. Also, it is proved that using fixed threshold for block type decision is sufficient and effective.

2.4 Proposed Mode Decision Scheme for H.264/AVC Encoder

This section takes full advantage of MB homogeneity detection to accelerate H.264/AVC intra and inter coding. Based on homogeneity judgment, 16x16 or 4x4 block size is appropriately selected for intra coding; Meanwhile, either the large blocks in {16x16, 16x8, 8x16} or sub-blocks in {8x8, 8x4, 4x8, 4x4} are chosen for inter coding. By using block typed decision, encoding complexity savings for the whole encoder is 43.7%-58.7%, with negligible loss in bit rate and PSNR.

2.4.1 Observations

As discussed in section 2.3.1, I4MB is suitable for images with significant details and I16MB prediction is well suited for smooth areas. If we can judge MB's homogeneity accurately we can safely skip unnecessary block sizes for intra coding. Similarly things happen for inter prediction. Figure 2.7 shows one example frame all from “News.cif”.

Table 2.5 Probability of encoding block sizes in News.cif

Blocks	16x16	16x8	8x16	P8
Probability	71.87%	4.26%	3.98%	19.89%

In this figure, the homogenous regions such as background, the suits, table and books on table, hairs of the two people are coded by 16x16 block, marked in red and green color. On the other hand, the people's moving eyes, mouths and shoulders are coded in smaller sizes. The different size of white boxes overlay on the image represents corresponding block modes determined by RDO. In this figure, the percentage of block sizes is shown in following Table 2.5.

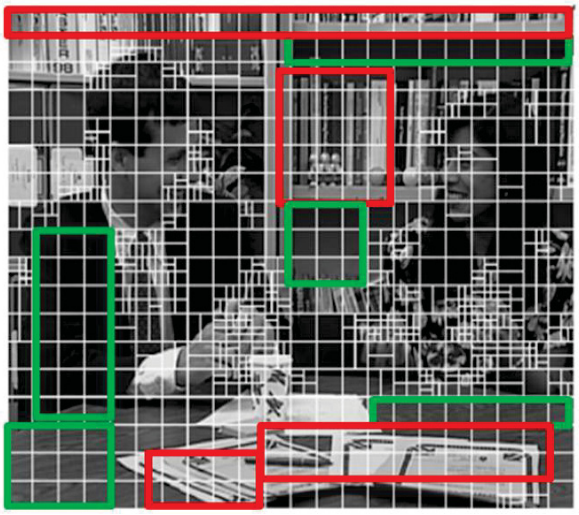


Figure 2.7 Block sizes selected by H.264 inter RDO

When QP value changes, the distribution of the optimal coding block size varies for As QP value becomes higher, the larger block sizes have more chances to be chosen as the optimal coding block. However, this trend does not conflict with the fact that homogeneous and motion-still area tends to be coded in large block sizes.

Based on the above observation, it is clear that if MB's homogeneity can be judged accurately, unnecessary blocks can be safely skipped for both intra and inter coding, without affecting the video quality. In this way, unneeded signal processing steps in motion estimation and RDO processes can be avoided to reduce encoding complexity.

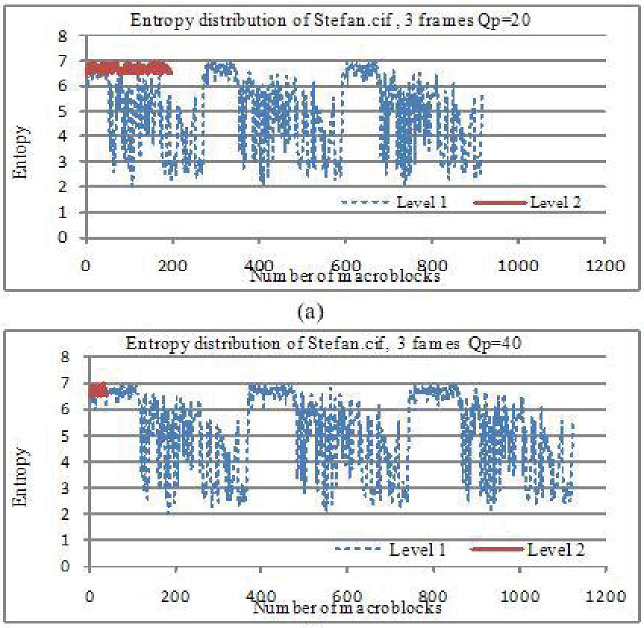
2.4.2 Mode Decision Scheme for Whole Encoder

This section introduces an overall mode decision scheme which can optimize both intra and inter coding using homogeneity detection. The mode decision scheme is consist of two main parts: (1) Intra Block Size Decision (Intra_BSD) algorithm is in charge of selecting the optimal coding block, using MB's entropy feature. (2) Inter block size decision (Inter_BSD) algorithm. It is responsible for separate blocks in level-1 {16x16, 16x8, 8x16} from blocks in level-2{8x8, 8x4, 4x8, 4x4}. And only one level will be chosen for inter coding, using homogeneity measurement based on MB's entropy feature. For intra prediction, our method is just identical with the proposal introduced in section 2.3.2. Next, optimization for inter part and their combination will be introduced.

As pointed out in previous sections: homogenous inter MBs and MBs with slight motions tend to use large block sizes; MBs containing different motions tends to be coded in size smaller than 8x8. Inspired by the 2 level tree structure defined in H.264, we only analyze the entropy distribution for block groups, namely level-1 blocks and level-2 blocks. Figure 2.8 demonstrates the entropy distribution of 3 P frames from "Stefan" with QP equals 20 and 40, respectively. It can be noticed very clearly that the majority of the entropy values of level-1 blocks lies below that of level-2 blocks. In other words, the entropy feature can be utilized to classify MBs to be level-1 blocks or not. According to the above observations, it is straightforward to use thresholds for deciding level-1 partition or level-2 partition based on entropy feature.

Similarly, the problem turns out to be classifying MBs into level-1 MB or level-2

MB based on entropy feature with an appropriate threshold. Take Figure 2.8 for explanation, it is observed that in most cases the entropy values for MBs coded by level-1 blocks range in [2, 7]; and those for MBs coded by level-2 blocks typically range in [6.5, 7]. Naturally, one threshold, if denoted as λ, is enough for classifying purpose. As a result, the best value is 6.5. Hence, the decision rule is like following:



(a)

(b)

Figure 2.8 (a) Entropy distortion of level-1 inter blocks of forman.cif, QP=20
(b) Entropy distribution of level-2 inter of foreman.cif, QP=40

- If entropy H is lower than 6.5, MB will be coded using large blocks {16x16,16x8,8x16};
- Otherwise, all blocks in level 1 an 2 will be used.

The encoding flow of proposed mode decision scheme can be explained as follows.

1) For one MB to be processed in inter coding. Calculate its entropy feature.
2) If above entropy value is less than 6.5, go to step 3; otherwise go to step 4.
3) If co-located MB is coded by SKIP mode, the best mode of current MB is inter 16x16 and go to step 6. Otherwise go to step 5

4) All inter modes will be active and encoder do all RD process for these modes. The one which gives minimum RD cost is the best inter mode. Go to step 6.

5) The best inter mode is 8x16/16x8. Go to step 6.

6) Now continue intra coding. If the entropy feature is less than 3.2, the best intra coding block size is I4MB, otherwise go to step 7.

7) If the entropy value is larger than 4.6, the best intra coding block size is I16MB, go to step 8. Otherwise go to step 8.

8) Decide the best mode for current MB.

2.4.3 Experiments and Comparison Results

Proposed entropy based MB homogeneity detection method is implemented into JM for block size decision in H.264/AVC. The test condition is listed in Table 2.6. Three measurement parameters, ΔPSNR, ΔBIT-RATE and ΔTIME calculated as [18], are used to evaluate coding performance.

Table 2.6 Test conditions for proposed overall mode decision scheme

JM Software Version	12.4
Profile	Main
# of tested frames	300
Tested frame format	QCIF-/CIF
Reference # for inter BSD	5
MV search Range	±16
MV resolution	1/4-pel
QP	28,32,36,40
Rate Distortion Optimization	on
Rate Control	off
GOP	IPPP
Entropy Coding	CABAC
Fast Motion Estimation	on

Proposed Mode Decision Scheme has been verified on a wide range of video sequences. Among tested videos, “foreman” and “container” cover only low spatial complexity and low motion. “Paris”, “silent” and “news” contain medium level spatial complexity and motion. “Mobile” and “Tempete” feature high spatial complexity and complicated motion activities. Thus, they can be classified into 3 classes.

1) Class A: Low spatial complexity and/or low motion
2) Class B: Medium level complexity and/or medium level motion
3) Class C: High spatial complexity and/or complex motion activities

Table 2.7 shows the performance in terms of PSNR, bitrate and coding time with IPPP using proposed Inter_BSC approach. Here, the above Intra_BSC method is not used for I frames and only the Inter_BSC method is utilized to select the appropriate level when P frames are encoded. In order to compare with [21] under the same test conditions, proposed Inter_BSC is implemented into JM encoder with fast motion estimation and Pan's fast intra mode decision algorithm in [29]. And the results of Inter_BSC in terms of ΔPSNR, ΔBIT and ΔTIME are against JM encoder with fast motion estimation and fast intra. It is observed that proposed approach can reduce the entire coding time by 43.14% on average with very little degradation in PSNR and bitrate, -0.05dB and 0.36%, respectively. Although the coding performance of proposed Inter_BSC is not always better than [21] for every tested sequence for every evaluating parameters(PSNR, BIT-RATE and TIME), our proposal has definitely shown consistent gain in coding speed for all videos with different characteristics, the least gain of 38.96% in *Container* and the most gain of 46.95% in *Stefan*. On the contrary, method in

Table 2.7 Performance of proposed Inter_BSC method and its comparison results with [21]

Sequence	Wu's method in [21]			Proposed Inter_BSC		
	ΔPSNR (dB)	Δ BIT (%)	ΔTIME (%)	ΔPSNR (dB)	Δ BIT (%)	ΔTIME (%)
Foreman	-0.06	1.28	-25.18	-0.04	0.44	-42.30
News	-0.06	1.18	-42.62	-0.05	0.19	-39.77
Container	-0.01	0.30	-36.25	-0.02	0.20	-38.96
Silence	-0.02	0.47	-45.16	-0.01	0.41	-43.57
Paris	-0.04	0.87	-31.90	-0.10	0.19	-45.39
Mobile	-0.005	0.13	-9.97	-0.08	0.98	-45.07
Stefan	-0.01	0.33	-17.37	-0.07	0.14	-46.95
Average	-0.03	0.65	-29.77	-0.05	0.36	-43.14

[21] leads to significant degradation in coding time for sequences with fast motions or complicated background, like *Stefan* and *Mobile,* respectively. Take “mobile” as an example: although it has a lot of small moving objects with complex content, the gain of coding time in our proposal is up to 45%, much higher than 9.97% in [21]. In addition, compared with [21], proposed approach is able to reduce an extra of 0.29% bitrate with only 0.02dB loss in PSNR in average. Compared with similar work of TSC in [28] which only gains 13% coding time reduction on average, proposed methods obviously can be a better competitor for H.264/AVC.

Since Intra coding mode is usually utilized in coding P and B frames, proposed Intra_BSC and Inter_BSC can work together as an entire inter mode decision algorithm for any IPPP videos. Thus, for each MB to be encoded the following procedure is followed:(1) If the MB is an I-frame MB, only the Intra_BSC method is activated and it is responsible to select either the I4MB or I16MB to encode the target MB;(2) If the MB is a P-frame MB, both the Intra_BSC and Inter_BSC method will be activated. The Intra_BSC method sees to select either I4MB or I16MB as final possible candidate to encode target MB; then, the Inter_BSC method is responsible to choose either Levl-1 blocks or Level-2 blocks as final possible candidates. Finally, the best coding blocks size will be the one that gives the minimum RD cost.

To compare with Liu’s work in [22], the same video sequences have been tested under same condition. Among tested video sequences, *Paris* is a sequence containing medium spatial complexity and medium motions. *News* is a sequence of low spatial detail that contains medium level changes in motion. *Silence* is a sequence of low spatial detail medium changes in the motion of the arms and head of the person in the sequence. *Foreman* is a sequence with medium changes in motion and contains dominant luminance changes. *Mobile* contains slow panning, zooming, a complex combination of horizontal and vertical motion and high spatial color details. *Coastguard* contains fast motions and camera panning. Table 2.8 lists the performances of PSNR, BIT and TIME for each sequences encoded with IPPP. Proposed inter mode decision algorithm can reduce the entire coding time by 52.05% on average with only negligible loss in terms of bitrate and coding time, -0.062 dB and 0.88%, respectively. Proposed method also shows consistent gain in encoding time for all videos with the least gain of 43.7% in *Salesman* and the most gain of 58.76% in *Grandma*. However, Liu's method shows an obvious degradation in encoding time for sequence with complex motions like *Mobile (*only 28.2%*)*. Compared with Liu's method, the proposed inter mode decision algorithm is faster in encoding speed and can reduce an extra 11.95% coding time averagely. Meanwhile, it achieves better performance in terms of average bitrate by saving an extra of 0.55% bitrate at almost the same PSNR.

Table 2.8 Performance of proposed mode decision scheme compared with [22]

Sequence	Liu's method in [22]			Proposed Inter mode decision method		
	ΔPSNR(dB)	ΔBT(%)	ΔTIME(%)	ΔPSNR(dB)	ΔBT(%)	ΔTIME(%)
Salesman	-0.080	1.70	-43.6	-0.013	0.62	-43.70
Grandma	-0.018	0.49	-45.7	-0.022	-0.33	-58.76
Stefan	-0.082	1.62	-33.0	-0.082	1.08	-53.12
News	-0.114	2.43	-49.2	-0.055	1.11	-57.39
Silence	-0.036	0.91	-49.6	-0.016	1.07	-52.25
Paris	-0.072	1.81	-41.0	-0.089	2.00	-49.43
Foreman	-0.049	1.48	-37.7	-0.112	1.22	-56.34
Coast	-0.022	0.64	-33.1	-0.055	0.19	-50.17
Mobile	-0.073	1.77	-28.2	-0.118	1.04	-47.31
Average	-0.061	1.43	-40.1	-0.062	0.88	-52.05

However, it can be seen from Table 2.4 and Table 2.7 that the integration of the two BSC approaches leads to a little worse coding performance. Take *Mobile* CIF sequence as an example, the PSNR values for Intra_BSC, Inter_BSC and their integration are 0.001dB, -0.08dB and -0.118dB, respectively. And the △bitrate for above three approaches are 0.05%, 0.98% and 1.04%, respectively. The encoding time reductions for these three methods are 34.17%, 45.07% and 47.31%, respectively. As a result, the coding performance of the integrated BSC method is a little worse than the addition of the encoding loss of the two methods. However, the encoding performance of integrated inter mode decision method is quite acceptable, which can be figured out from the Rate-Distortion curves in Figure 2.9. The performance of the Inter_BSC algorithm and its integration with Intra_BSC method is compared with JM 12.4 encoder for videos with different spatial/motion characteristics.

Figure 2.9 shows Rate-Distortion curves of three representative video sequences: *Akiyo*, *Paris* and *Stefan*. *Akiyo* stands for low spatial/temporal videos and *Paris* is a sequence of medium spatial complexity and medium motion. *Stefan* contains panning motions and has distinct fast changes in motion with complicated background. For *Akiyo* sequence shown in (a), proposed Inter_BSC method (shown in red) achieves almost identical RD performance as that provided by JM model (shown in blue) for the two curves are overlapped. Moreover, the RD performances of the proposed inter mode decision algorithm (shown in green) just decreases very little compared with that of JM model. For *Paris* sequences shown in (b), it can seen that the Inter mode decision algorithm(shown in green) achieves almost identical coding performance with that provided by Inter_BSC algorithm (shown in red) . In addition, the above 2 algorithms achieves almost the same coding performance when bitrate is below 2000 Kbps. Although the coding performances of proposed algorithms are a little decreased when bitrate is larger than 2000Kbps, it still achieves very similar coding efficiency compared with JM software. For *Stefan* sequence shown in (c), almost all the RD curves are identically overlapped when bitrate is below 4000 Kbps. Although the coding performance of proposed algorithms is a little decreased when bitrate is larger than 4000Kbps, it still achieves similar coding efficiency compared with that of JM model.

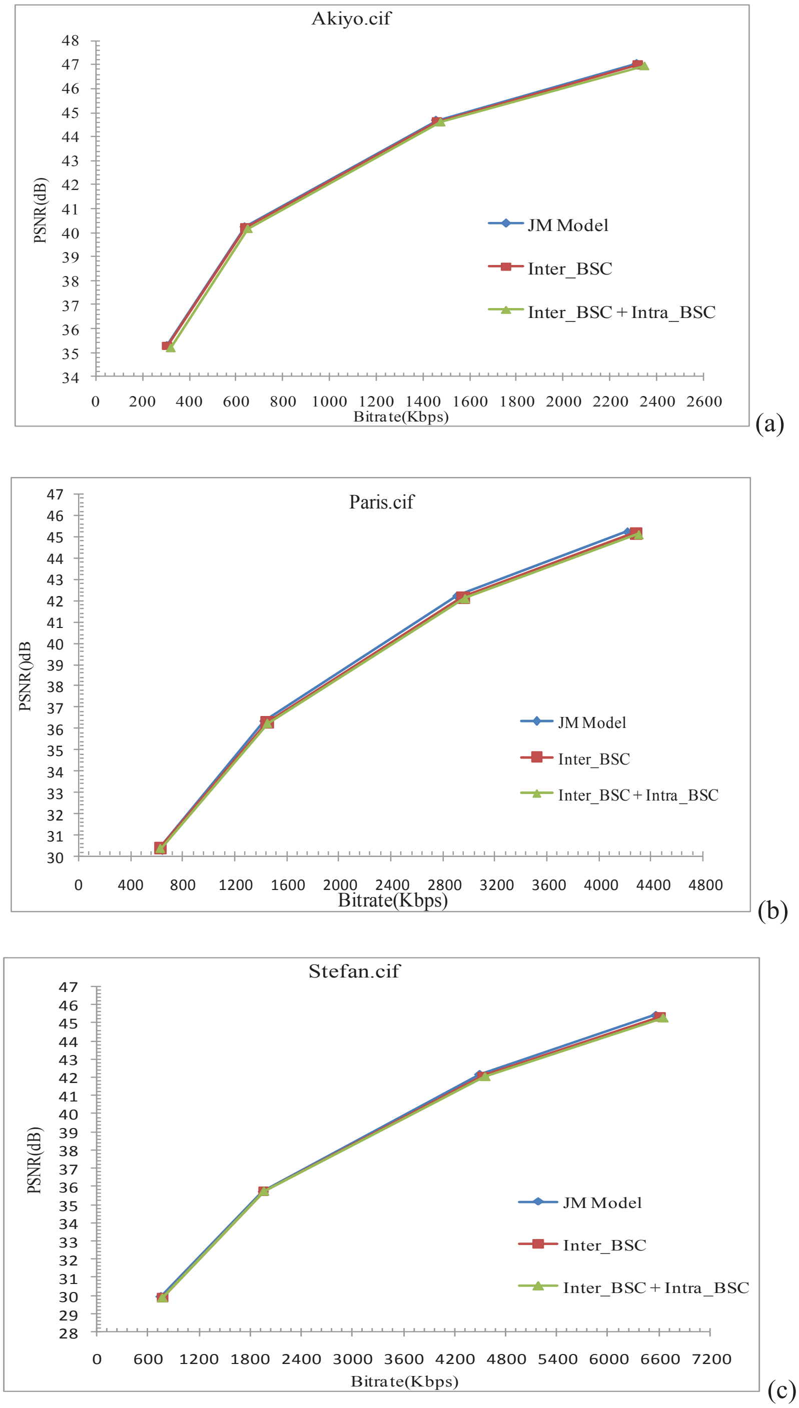


Figure 2.9 Rate distortion curve of sequences (a) Akiyo (b) Paris (c) Stefan

2.5 Conclusion

This chapter firstly proposes an efficient fast intra mode decision algorithm. Prewitt operator is used to extract each block’s two most dominant edge directions. Based on these two directions, only a few modes are chosen as candidates. In this way, computational load caused by RDO can be reduced remarkably. Experimental results show that more than 70% computation reduction can be achieved, with little loss of visual quality. Moreover, only addition and comparison operations are used, which makes proposed algorithm friendly enough to be implemented in hardware.

Then, it proposes a novel entropy feature based MB homogeneity measurement method and applies it for inter-/intra block sizes decision for H.264/AVC. Proposed entropy feature is extracted directly from raw pixels contained in each MB which enables proposed block size decision method to explore the nature of input videos. The proposed method can classify I4MB and I16MB for intra coding and level-1 and level-2 blocks for inter coding accurately and with fast speed. In this way, coding complexity is greatly reduced for all videos with different motion and spatial characteristics: encoding complexity for intra coding alone can be reduced by 31%-34% and time savings for inter mode decision is 43.7%-58.7%, both with very negligible loss in terms of bitrate and PSNR. The above performance is verified on a different video sequences which contain a wide range of texture and motions.

Compared with the most latest work, Liu’s method (IEEE CVST 2009) [22], the proposed inter mode decision algorithm is not only faster by 11.95% in encoding speed but also can reduce an extra of 0.55% bitrate with PSNR value decreased by 0.001 dB. This make proposed approach to be a favorable accelerator for a real time H.264/AVC encoder.

3 CONTENT ADAPTIVE HIERARCHICAL DECISION OF VARIABLE BLOCK SIZES IN HEVC

3.1 Background and Previous Proposals

This section introduces the basic background and previous works related to variable block sized hybrid prediction using a quadtree structure.

3.1.1 Hierarchical Structure of Variable Block Sizes in HEVC

As explained in section 1.2.1, HEVC adopts a quadtree structure to organize coding unit (CU), prediction unit (PU) and transform unit (TU), each are labeled with depth, to

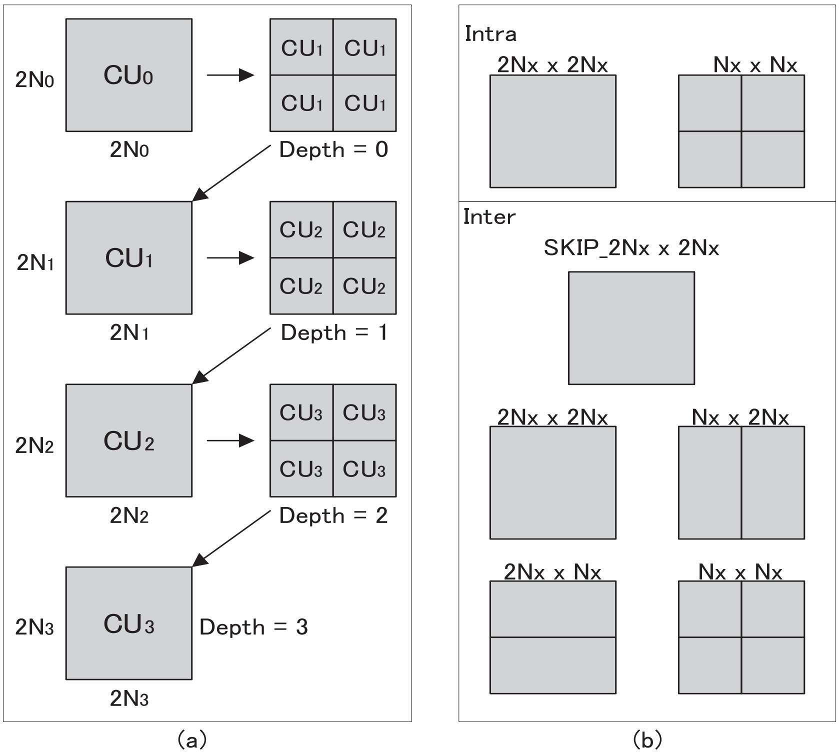


Figure 3.1 (a) Recursive quadtree-based partition for an LCU.
(b) Available prediction block sizes for a $2N_x$x$2N_x$ CU at depth x.

partition image frame, carry out motion estimation/compensation and encode residual data. Figure 3.1 illustrates the way of recursive partition with the quadtree structure. A 64x64 image block called largest coding unit (LCU) is recursively split and hybrid-predicted by variable-sized prediction blocks. Figure 3.1(a) shows an example of recursive partition which forms a 4-depth quadtree; for a $2N_x x2N_x$ coding block (CU), supported prediction blocks are indicated in Figure 3.1(b).

Quadtree-based prediction makes motion representation more accurate and greatly improves compression efficiency. Nevertheless, our experimental results by encoding sequences with representative features in different resolutions indicate that, huge computational complexity, about 75-80% of the total complexity of an HEVC encoder, is required to process such a wide-range prediction blocks summarized in Table 3.1.

Table 3.1 Coding unit sizes and prediction block (PU) sizes in HEVC.

Depth	CU	Inter PU	Intra PU	Transform block
0	64x64	64x64,64x32,32x64	64x64	32x32, 16x16
1	32x32	32x32,32x16,16x32	32x32	32x32,16x16,8x8
2	16x16	16x16,16x8,8x16	16x16	16x16,8x8,4x4
3	8x8	8x8,8x4,4x8,4x4	8x8,4x4	8x8,4x4

Table 3.2 Number of supported intra modes according to PU size

PU size	4x4	8x8	16x16	32x32	64x64
Number of modes	17	34	34	34	5

3.1.2 Directional Intra Prediction in HEVC

In intra coding, except for the variable block sizes for better spatial adaptability, directional modes are also enhanced to maximally 33 and one DC mode for better compression efficiency. The total number of available prediction modes is dependent on the size of the corresponding PU, as shown in Table 3.2. Based on the experience from H.264/AVC, the increment of intra prediction modes will cause a great burden to throughput since for each LCU, 256x4 + 34x(64+16+4) + 5 = 3885 times of RDO computation has to be finished before coding the next LCU. Therefore, HM1.0 adopts a rough mode decision to boosting the process of mode decision. For each CU, the rough decision calculates hadamard cost between original image block and its predicted block using all available prediction direction (modes), only a limited number of modes which gives less cost are selected for RDO decision. The last row of Table 3.2 illustrates number of modes after rough decision.

3.1.3 Previous Proposals

Work [38] further decreases the selected modes decided by rough decision to 5, 5, 2, 2 and 1 for 4x4, 8x8, 16x16, 32x32 and 64x64 PU, respectively. Moreover, it adds one more prediction mode called most probable mode (MPM) for each PU. The MPM originally is from H.264 which represents the fact the neighboring PU has high probability to have similar texture direction. Formally, MPM is defined as

$$\begin{aligned} &P(M_{cur} \middle| (M_A, M_B)) \\ &= P(Mode_{cur} = \min(M_A, M_B)) \middle| (Mode_A = M_A, Mode_B = M_B) \end{aligned} \tag{3.1}$$

where, $Mode_{cur}$, $Mode_A$ *and* $Mode_B$ represent the optimal intra mode of current block and its two neighboring two blocks located on left and top of current blocks individually. By using above two approaches, [38] reduces intra prediction time by 14% and 20% for All-Intra High Efficiency and All-Intra Low Complexity configures, respectively. At the same time, BD-rates is creased by 0.06% for the later configuration. This proposal has been adopted in HM 2.0 becomes standardarlized process for intra prediction.

Part of coding depths is skipped by comparing utilization ratio of contiguous coding depths in [39]. The specific rule of skipping is shown by Eq. (3.2)

$$Th2 < \frac{Num_{depth(x)}}{4 \times Num_{depth(x+1)}} < Th1 \qquad (3.2)$$

where, *depth(x)* = *x* for x equal to 0, 1, 2 and 3, which stands for a coding depth. $Num_{depth(x)}$ by its name is the number of CU in depth x. By using the right hand term in Eq.(3.2), if the number of CU in depth x is much less than that in depth (x+1), CU in depth x will not be applied for prediction. The same depth will be utilized again when the left term of Eq. (3.2) is satisfied. Essentially, [39] calculates the adoption ration of continuous depths and the one with lower ratio will be skipped. Thus, the skipping is based on relative comparison and global utilization ratio is not considered. With many irregular motions or complex background, coding gain of [39] will be greatly degraded.

It might be possible to reuse the ideas in [21] and [40] for HEVC in the meaning of choosing proper prediction blocks in depth 2 and 3. However, problem remains unsolved for selecting blocks in depth 0 and 1, not to mention that these two depths require even more computations at ME and residual encoding stages [41].

JCTVC-F092 [23] takes advantage of the observation that if the best mode of current CU is SKIP mode, motion estimation and compensation for sub-tree is a waste, and proposes an early termination (ECU) technique to disable further CU partitions. Experiments of [23] show HEVC encoder speeded up by 40% with BD loss about 0.6%. ECU is effective for videos with stationary motions, homogenous texture and high QP coding. However, acceleration by ECU becomes limited when SKIP mode happens infrequently for low QP and for videos with complicated texture or motion.

3.2 Proposed Content Adaptable CU Decision for Intra Prediction

This section brings forward a two-stage prediction unit size decision algorithm to speed up the original intra coding in HEVC. The proposed algorithm can speed up original HEVC intra coding by a factor of up to 73.7% and by averagely 44.91% for 4kx2k video sequences. Meanwhile, the peak signal-to-noise ratio degradation is less than 0.04dB and bit-rate stays almost the same.

3.2.1 Observation

The sum of Sobel amplitude is measured as MB's smoothness to distinguish 16x16 and 4x4 blocks [19]. Another approach uses entropy of MB for the same purpose [42]. Reference [20] extracts DC energy of 8x8 block to select proper block types for intra coding of high profile. However, when the problem comes to decision of five different PU sizes, the more important issue is threshold setting. Our simulations demonstrate that for the same PU level and the same QP, the texture complexity ranges of different sequences vary greatly. In addition, for the same PU level at different QP, the texture complexity ranges of the same sequence are also different. Therefore, each sequence should have its optimal thresholds which can adapt to both video content and coding parameters, in order to get the best trade-off between quality and complexity reduction. In this sense, previous methods can hardly be extended to HEVC since thresholds in these methods are fixed value or can only adapt to QP.

3.2.2 Adaptable CU Decision Algorithm

It is already proved that homogenous image block tends to be coded by large blocks and complex ones usually are coded by small blocks [19] [20] and [42]. Complex image block tends to have significant dissimilarity in intensity in pixel domain, and dissimilarity can be detected easily by variance in statistics. In order to reduce computational overhead and get rid of noises, the original 64x64 luma LCU is down-sampled to 16x16 block. Each of the down-sampled pixels' intensity *P(i, j)* is the average intensity of four pixels in a none-overlapped 4x4 block in original luma LCU. After down-sampling, the complexity of the original LCU can be extracted as its variance. Then, variance is further normalized by its maximum value. Note that, complexity of the four 8x8 blocks in down-sampled 16x16 block are also calculated, in order to filter out useless PUs for original 32x32 blocks.

Threshold setting is crucial for PU decision. By simulations on various kinds of videos, it is found that the complexity ranges of LCU coded by the same PU size varies and the optimal threshold for each video depends on video content itself. Hence, content adaptive threshold setting scheme is brought forward for the pre-stage filtering algorithm. For each LCU in the 1st frame, its complexity is calculated before intra coding. After the encoding of each LCU, the minimal PU size used to encode the LCU is calculated. According to the minimal PU size, the texture complexity of LCU is buffered into one of the following buffers, *DSDC_buf_64_32*, *DSDC_buf_16* and *DSDC_buf_8_4*. *DSDC_buf_64_32* is used to save complexities of LCU which is encoded by one 64x64 or four 32x32 blocks. *DSDC_buf_16* is used to store complexities of LCU the minimal PU of which is 16x16 block. *DSDC_buf_8_4* is used in the same manner. When the 1st frame is finished, ranges (minimal and maximum complexity) of {64x64, 32x32}, {16x16} and {8x8, 4x4} are computed, note as [a0, b0], [a1, b1] and [a2, b2], respectively. Next, dynamic thresholds *Thre_0* and *Thre_1* are calculated as following pseudo code.

```
If (a1-a0 <0.05)        Thre_0 = a2; skip {8x8, 4x4} for LCU
else                    Thre_0 = a1; skip {16x16, 8x8, 4x4} for LCU
If (b1-b0 <0.05)        Thre_1 = b0;
else                    Thre_1 = b1;
```

Note that the 1st frame is encoded in normal way and successive frames are coded by proposed algorithm. After every 50 frames, the threshold setting process executes once again, to make sure the threshold are updated as video content changes.

To reduce unnecessary PU candidates, if the complexity of LCU is high, large PUs are filtered out; on the contrary, if the extracted complexity is low, small units like {8x8, 4x4} are excluded. In addition, the complexities of the four 8x8 blocks in each down-sampled 16x16 block are computed, to filter out improper PUs for original 32x32 CUs. The processing flow of filtering out unnecessary PUs is shown in Figure 3.2.

1) Down-sample the 64x64 LCU to 16x16 block.

2) Calculate texture complexity of the 16x16 block using (3), noted as *DSDC_16x16*.

3) If *DSDC_16x16* is not larger than threshold *Thre_0*, small PUs like {8x8, 4x4} are filtered out. For some sequences, {16x16} is also filtered out (shown in the pseudo code in sub-section B). Go to step 8. Otherwise, go to step 4.

4) If *DSDC_16x16* is larger than *Thre_1*, large units {64x64, 32x32} are skipped and go to step 8. Otherwise, go to step 5.

5) Calculate the four complexities of 8x8 sub-blocks in down-sampled 16x16 block and compute the maximum and minimum values, noted as *DC_8x8_min* and *DC_8x8_max*. If *DC_8x8_min* is less than *Thre_0*, skip {8x8, 4x4} which gives the minimal complexity and go to step 8. Otherwise, go to step 6.

6) If *DC_8x8_max* is larger than *Thre_1*, skip 64x64 for the original LCU and 32x32 for the 32x32 block which gives the maximum complexity. Go to step 8. Otherwise, go to step 7.

7) Perform during stage skipping process and then go to step 8.

8) Encode the LCU with remaining PUs.

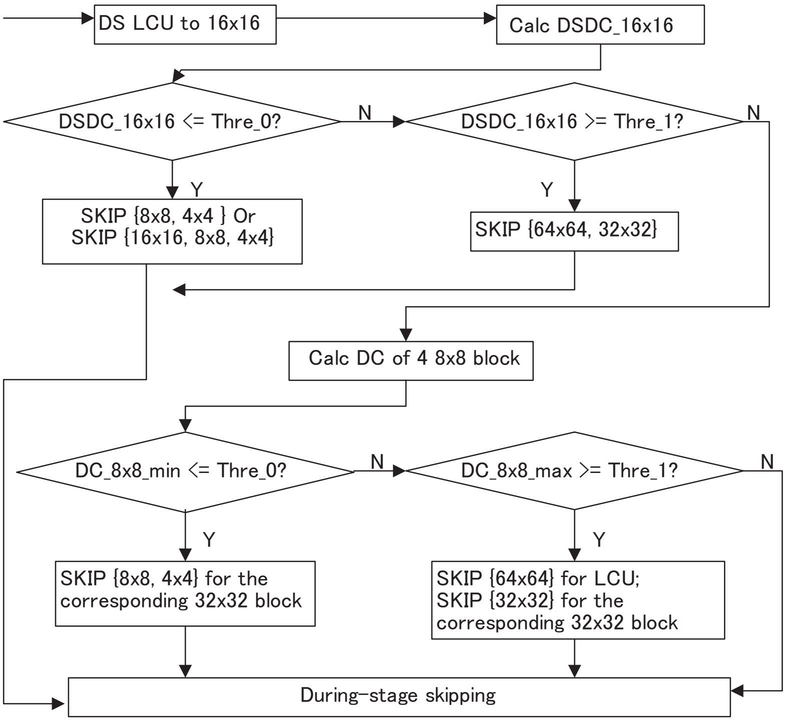


Figure 3.2 The processing flow of overall PU decision algorithm

Like the MPM mode in intra prediction in H.264/AVC, neighboring blocks have lots of similarities and it is possible that their optimal coding depth is also the same. Thus, we can skip small PUs when encoding the last 32x32 block in a LCU during intra coding process, by utilizing the PU sizes of encoded upper-left, upper and left 32x32 blocks. It is known that intra coding utilizes encoded upper and left blocks to reduce redundancy in current block. The UL, U and L blocks inside of the same LCU shown in Figure 3.3, have high probability to have similar contents. Therefore, it is with high possibility that their optimal PU sizes tend to be the same. The proposed during-stage skipping scheme works as following.

1) Encode the first three 32x32 blocks, UL, U and L using the proposed pre-stage filtering method.
2) Calculate the minimum PU sizes of UL, U and L.
3) If the minimum PU sizes for the three blocks are the same, note as *M*x*M*, skip PU

size smaller than *MxM* for the current (noted as Cur. in Figure 3.3) 32x32 block.

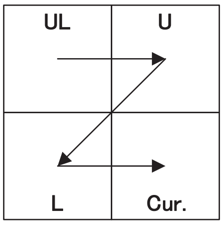


Figure 3.3 During-stage skipping for the last 32x32 block

3.2.3 Experimental Results

The proposed algorithms are combined together and implemented into the original HM 4.0 test mode [43]. Quantization parameters are 22, 27, 32 and 37. Number of frames tested is 100 for 4kx2k sequences and 150 for 1920x1080p sequences. Table 3.3 lists the comparison results of proposed algorithm against original intra coding for 4kx2k sequences, using Intra Only High-Efficiency (HE) and Low-Complexity (LC) configurations [44], respectively. Evaluation parameters are *ΔPSNR*, *Δbit-rate* and *ΔEncTime*. It can be observed that the decrease in PSNR for all sequences is less than 0.01dB in HE configuration and less than 0.04dB in LC configuration. Bit-rate

Table 3.3 Encoding results comparison of our proposal against HM 4.0, tested on 4kx2k videos

Sequence	High Efficiency		
	ΔPSNR(dB)	ΔBit-rate(%)	ΔEncTime(%)
ducks_take_off	-0.009	-0.51	-14.97
crowd_run	0	0.02	-22.69
into_tree	-0.008	0.41	-39.43
old_town	-0.001	0.07	-40.69
park_joy	-0.001	0.40	-35.76
Avg_HE	**-0.0038**	**0.0007**	**-30.71**
Sequence	Low Complexity		
	ΔPSNR(dB)	ΔBit-rate(%)	ΔEncTime(%)
ducks_take_off	-0.033	0.007	-71.32
crowd_run	-0.022	-0.068	-35.01
into_tree	-0.008	-0.048	-65.11
old_town	-0.038	-0.153	-73.70
park_joy	-0.039	-0.433	-50.35
Avg_LC	**-0.028**	**-0.14**	**-59.10**
Total Average	**-0.016**	**-0.069**	**-44.91**

increment in HE configuration is little, with worst case of 0.41% increase and best case of 0.51% decrease. In LC configuration, bit-rate is even improved by 0.14% averagely. Meanwhile, coding complexity reduction is with the least gain of 14.97% for *ducks_take_off* in HE configuration and the most gain of 73.7% for sequence *old_town* in LC configuration. The average encoding complexity reduction for both configurations is 44.91% for all tested sequences.

Table 3.4 shows the comparison results using HE configuration only tested on 1920x1080p sequences, which are specified in [46] as common test videos consisting of different characteristics of video contents. It is observed that PSNR loss is less than 0.04dB for all sequences and bit-rate degradation is negligible, with best case of 0.17% increment and worst case of 0.78% increment. Average time reduction is 28.8%.

Table 3.4 Encoding results comparison of our proposal against HM 4.0, tested on 1080p videos

Sequence	ΔPSNR (dB)	ΔBit-rate (%)	ΔEncTime (%)
Kimono1	-0.034	0.17%	-27.91%
Cactus	0	0.45%	-20.74%
BQTerrace	-0.029	0.23%	-33.53%
ParkScene	0.002	0.71%	-30.43%
BasketballDrive	-0.006	0.78%	-31.39%
Average	**-0.013**	**0.47%**	**28.8%**

RD performance comparison against original intra coding is given in Figure 3.4. It can be seen that the two RD curves of proposed algorithm tested on *crowd_run* and *BaseketballDrive* sequences are identically overlapped with that using original intra coding process of HEVC. That is to say there is hardly coding performance loss using proposed algorithm. As to the effectiveness of proposed two methods, detailed number is not shown here for limited page length. Roughly, the pre-stage filtering saves more than half of the total complexity. The during-stage method is able to reduce about one-third of the total computational complexity. Proposed methods can also be used to speed up inter coding like lowdelay-P configurations.

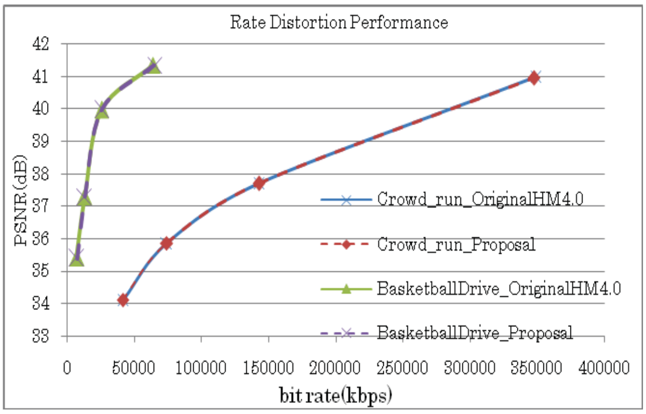


Figure 3.4 Rate Distortion performance compared with original HM 4.0

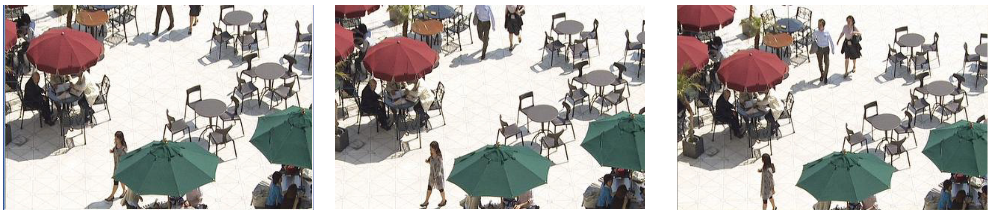
3.3 Proposed Content Adaptive Hierarchical Decision Scheme

This section proposes an adaptive scheme which fully exploits temporal, spatial and transform-domain features to reduce computational complexity in quadtree-structured prediction. By utilizing the three features, all videos with various characteristics can be dramatically accelerated. Experiments on plenty of video sequences show that original HEVC encoding is speeded up by up to 61.9% for 4kx2k videos with trivial degradation in PSNR and bit-rate.

3.3.1 Frame Level Decision Using Statistical features

This section describes frame-level skipping method which exploits temporal correlation for acceleration. Utilization ratio of each coding depth is analyzed using a start frame to skip rarely adopted coding depths for next several frames.

Consecutive frames usually have high similarity in both motion and texture. Figure 3.5 shows that both background and moving foreground in the 1st, 49th, 99th frames of BQSquare are with extremely high similarity. This observation holds true especially for sequences captured at high frame rate. Except for similarity between frames, coding parameters for continuous frames stay the same. Therefore, the optimal coding blocks used to encode certain regions in one frame tend to be selected again for similar regions that appear in next several frames. Consequently, utilization ratio of each depth tends to be stable among consecutive frames. For this reason, by analyzing utilization ratio of each depth, rarely used coding depth can be skipped for next frames.



(a) The first frame (b) 49th frame (c) 99th frame

Figure 3.5 Similarities between consecutive frames in BQSquare.

The left-hand term of Eq. (3.3) is used to calculate utilization ratio of each depth.

$$\frac{Num_{depth(i)}}{\sum Num_{depth(i)}} \le T, \ i = 0 \sim 3 \tag{3.3}$$

Where *depth(i)* standards for depth *i* (*i*=0, 1, 2, 3) and $Num_{depth(i)}$ represents the total number of LCUs whose depth is *i*. Here, the depth of an LCU is defined as the depth of the smallest coding block that is used to encode the LCU. *T* is a pre-defined threshold ranging from 0 to 1.0. Through intensive experiments on multiple sequences, *T* is set as

$$\begin{cases} T = 2\%, \ i = 0, 1, 2 \\ \quad T = 5\%, \ i = 3 \end{cases} \tag{3.4}$$

It is noted that *T* for depth 3 is larger than that for other depths. The reason is that LCUs even coded by a few depth-3 blocks are treated as depth-3 LCUs. As a result, the number of depth-3 LCUs is usually larger than the number of LCUs in other depth. Hence, a larger T is set for depth 3 for more coding complexity reduction.

After the first P frame is encoded as normal, utilization ratio of each depth is calculated using the left-hand term in Eq. (3.3). For any depth, if Eq. (3.3) is satisfied, all prediction blocks in that depth will be disabled for frames captured within one second. Calculation of utilization ratios of all four depths and comparison in Eq. (3.3) will be re-performed after encoding frames in every one second, in order to take scene changes into consideration.

3.3.2 LCU Level Content Adaptable Decision Using Spatial Features

This section introduces TC-based skipping which makes use of texture homogeneity in spatial domain to filter out useless coding blocks for each LCU. A novel threshold setting scheme is also proposed to make filtering adaptable to videos. Processing flow of TC-based skipping is given in the last subsection.

Large coding block is more efficient for homogeneous image areas and big moving objects. Small coding block benefits irregular motions and complex image region. In natural video sequences, homogenous and complex areas always exist [21]. So, texture complexity analysis is a useful technique for selecting proper prediction blocks [21] [40].

In general, complex block has significant dissimilarity in pixel intensity and dissimilarity can be detected by variance. To get rid of random noise, a 64x64 luma LCU (If IBDI is not 8-bit, an LCU is recovered to 8-bit) is firstly down-sampled to a block with 16x16 pixels. In down-sampling process, an LCU is spatially partitioned into 256 equal-sized blocks and each block is with size of 4x4. Next, the LCU is down-sampled by a factor of 4 horizontally and vertically by replacing each 4x4 samples with their average intensity. Variance of a down-sampled LCU can be extracted as

$$E_{\text{var}} = \sum_{i=0}^{i=15} \sum_{j=0}^{j=15} [P(i,j) - \tfrac{1}{256} \times (\sum_{i=0}^{i=15} \sum_{j=0}^{j=15} P(i,j))]^2 \tag{3.5}$$

where *P(i, j)* denotes pixel intensity at position *(i,j)*. The possible maximum value of E_{var} is the variance of a 16x16 checkerboard with 256 pixels of alternating 0 and 255, computed as

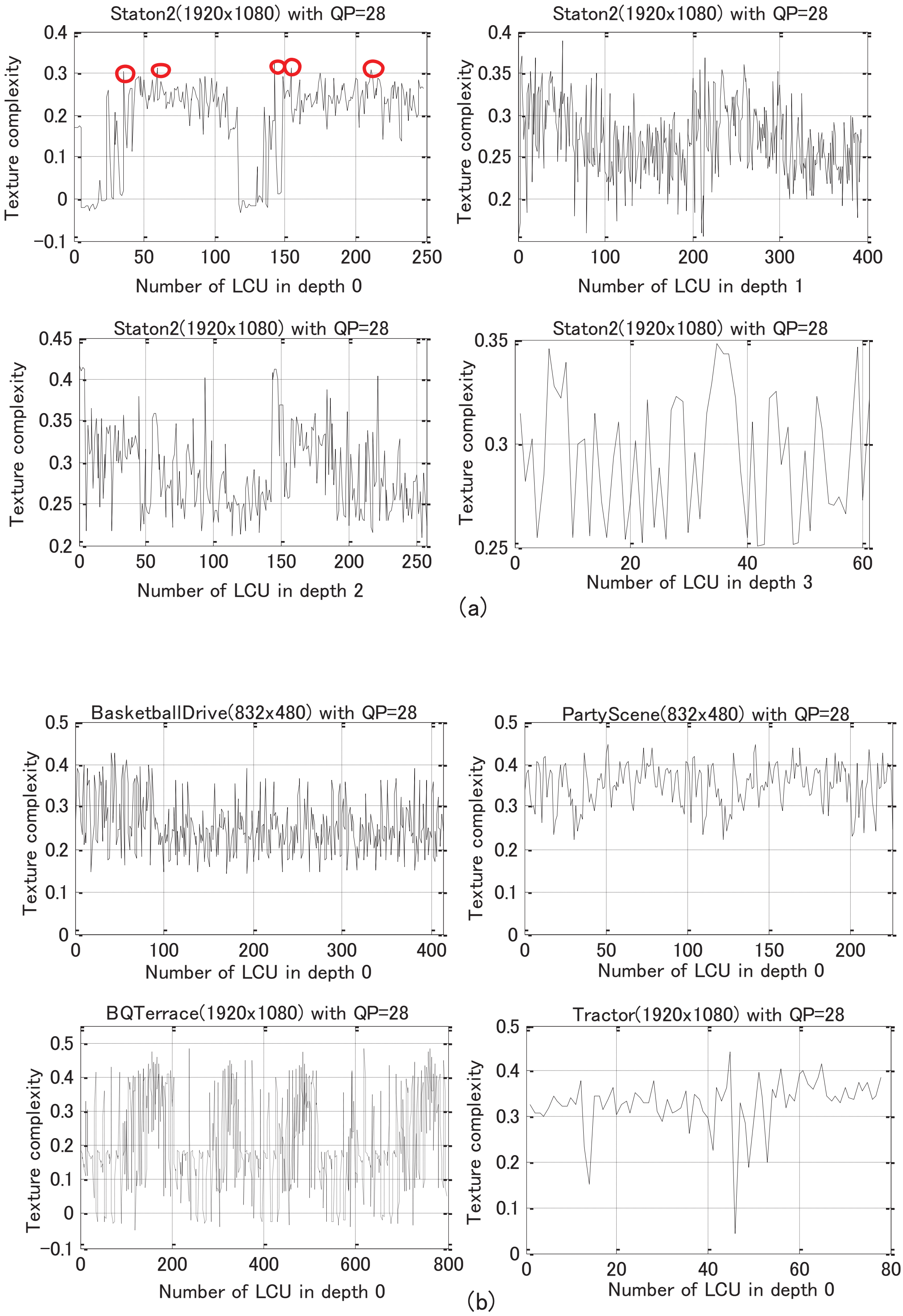
$$E_{\max} = \frac{1}{2} \sum_{0}^{15} \sum_{0}^{15} [(255^2 + 0^2) - \frac{1}{2} * (255 + 0)^2] \tag{3.6}$$

Texture complexity (TC) of an LCU is expressed by a normalized value defined as

$$TC = \frac{\log(E_{\text{var}})}{\log(E_{\max})} \tag{3.7}$$

Using Eq. (3.7), TC values of LCUs in several frames can be easily calculated and plotted. Figure 3.6 (a) shows TC distributions of LCUs in depth 0, 1, 2 and 3 respectively. To get the TC distributions, 2 frames with resolution of 1920x1080 from “Station2” are encoded by original HM 4.0 with QP equal to 28. As a result, number of LCUs in depth 0, 1, 2, and 3 are 248, 394, 257 and 61 respectively. It is observed that TC ranges for different depths are only partially overlapped. It indicates coding unit skipping is possible for LCUs in none-overlapped parts. For example, LCU with TC value smaller than 0.15 is coded by only depth-0 blocks. In this case, coding units in other three depths can be discarded. Figure 3.6 (b) shows TC distributions of depth-0 LCUs using four different sequences. It is noted that the four TC ranges differ from each other. The same phenomenon has been observed for other three coding depths. Conclusion from observations is that coding block size decision using fixed threshold is not appropriate to encode videos with different contents. In other words, thresholds used to choose coding units in HEVC must be adaptable to characteristics of videos.

We also do investigation on TC distributions of a same depth using multiple sequences with different resolution and features in texture. Any depth can be analyzed and we take



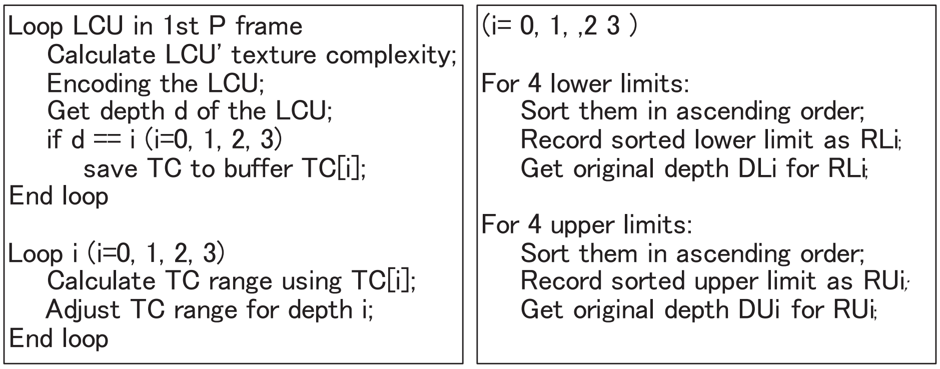
(a)

(b)

Figure 3.6 (a) TC distributions of four depths on the same sequece. (b) TC distributions of depth-0 LCUs on four different sequences.

depth 0 as an example. To get the TC distributions, 10 frames with resolution of 832x480 from *BasketballDrive* and *PartyScene*, 5 frames with resolution of 1920x1080 from *BQTerrace* and Tractor are encoded by original HM 4.0 at QP is 28. As a result, 413, 225, 797 and 78 LCUs in depth 0 in the four sequences are analyzed. Figure 3.6 (b) shows the TC distributions. It is noticed that TC range depends on texture of input sequence. TC range can also be affected by QP. Although QP has no impact on calculation of TC in Eq. (3.7), it affects the depth of an LCU, which will in turn affect TC range of a coding depth. Therefore, the optimal coding blocks of an LCU depend on texture complexity and QP. To take these factors into consideration and set 8 thresholds properly, an adaptive threshold setting scheme is proposed.

The main idea is to calculate TC ranges of four depths using a start frame and reuse these ranges as thresholds for skipping none-contributable blocks for next several frames. Pseudo code to calculate TC ranges for each depth is given in Figure 3.7 (a). When encoding of the first P frame is finished, TC values are gathered into four depths. By sorting all TC values in each depth, TC range of this depth can be expressed by the minimal and maximal TC values, in the form of [min TC, max TC].



```
Loop LCU in 1st P frame
    Calculate LCU' texture complexity;
    Encoding the LCU;
    Get depth d of the LCU;
    if d == i (i=0, 1, 2, 3)
        save TC to buffer TC[i];
End loop

Loop i (i=0, 1, 2, 3)
    Calculate TC range using TC[i];
    Adjust TC range for depth i;
End loop
```

(a)

```
(i= 0, 1, ,2 3 )

For 4 lower limits:
    Sort them in ascending order;
    Record sorted lower limit as RLi;
    Get original depth DLi for RLi;

For 4 upper limits:
    Sort them in ascending order;
    Record sorted upper limit as RUi;
    Get original depth DUi for RUi;
```

(b)

Figure 3.7 (a) Pseudo code of calculating TC ranges. (b) Pseudo code of sorting TC lower/upper limits.

Threshold setting is always the trade-off between coding performance and time reduction. Instead of using [min TC, max TC] directly, an adjusted range is used. Take the first figure in Figure 3.6 (a) as an explanation. While the majority of TC values is below 0.3, five TC values marked in red circle are larger than 0.3. If these “TC noise” are excluded, depth-0 blocks can be disabled for an extra number of LCUs the TC values of which are in [0.3, 0.32]. So, TC noise need to be got rid of for more complexity reduction. As a result, TC range of each depth will be slightly adjusted.

For adjustment, histogram of TC values can be used. The range of [-1.0, 1.0] is divided into 41 bins with interval equal to 0.05. Previously gathered TC values in each depth are then mapped to corresponding bins. After mapping, histogram of each depth is generated and cumulative histogram is utilized as follows to get new ranges.

$$\sum_{i=-20}^{i=20} density_{[B_i, B_{i+1}]} \geq TH \tag{3.8}$$

In Eq. (3.8), [B_i, B_{i+1}] denotes the bin of $[0.05\times i, 0.05\times(i+1)]$, and *density* $_{[Bi, Bi+1]}$ is the frequency that TC values in [B_i, B_{i+1}] appear. *TH* is defined as 2.5% and 97.5% for lower and upper limits, respectively. These two values ensure a typical confidence interval of 95% in applied practice. The first value of *i* which satisfies Eq. (3.8) is used to compute adjusted TC range according to Eq. (3.9).

$$\begin{cases} TH = 2.5\%, RL = i \times 0.05 \\ TH = 97.5\%, RU = i \times 0.05 \end{cases} \tag{3.9}$$

RL and *RU* are lower and upper limit of adjusted range, respectively. By performing the operations in Eq. (3.8) and (3.9) four times, new TC ranges of all four depths can be obtained. Next, the four lower limits and four upper limits are sorted in ascending order and their original depths are recorded. The pseudo code for this step is given in Figure 3.7 (b). Sorted 8 values are the thresholds which will be used for depth(s) skipping.

Proposed thresholds setting method completely adapts to input videos and coding parameters. The 8 thresholds are dynamically changed for different combinations of input videos and coding parameters. The threshold setting process is executed after encoding frames in every one second, to make sure thresholds are refreshed.

Suppose sorted TC ranges satisfy RL_0<=RL_1<= RL_2<=RL_3 and RU_0<= RU_1<= RU_2<= RU_3, and their original depths are the appeared subscripts. Processing flow of coding unit skipping using TC measurement and proposed threshold setting method is summarized in Figure 3.8.

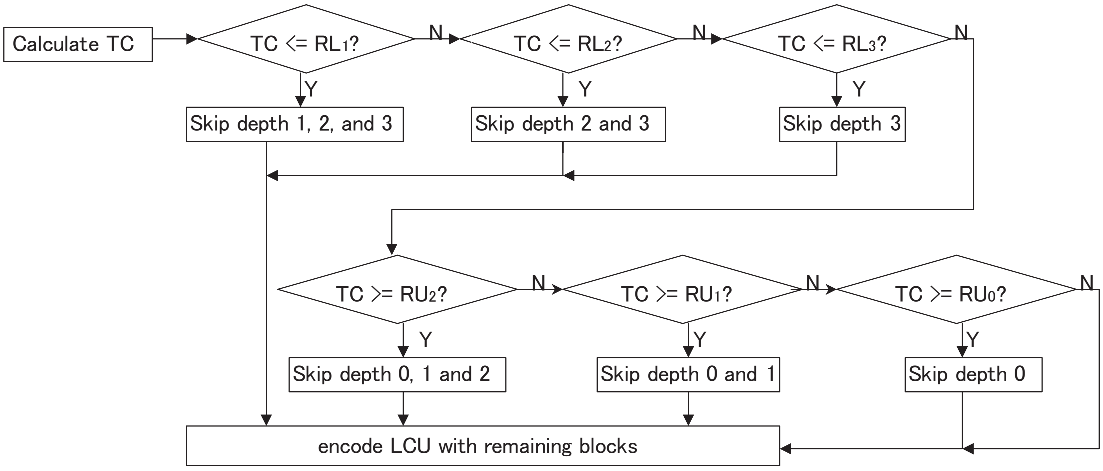


Figure 3.8 Encoding flow of TC-based coding unit skipping for an LCU.

For each LCU in the second and later P frames, its TC feature is firstly calculated using original luminance video signal according to Eq. (3.7). Then, the TC value is compared with the 8 thresholds respectively. If the calculated TC value is no larger than RL_i (i=0, 1, 2, 3), all coding units in depth *i* to depth 3 are skipped; otherwise, if TC is larger than RU_i, coding blocks in depth 0 to depth *i* are skipped. Proposed method can skip up to 3 depths for some LCUs.

3.3.3 CU Filtering Using Transform features

This section uses SAQC in transform-domain to prune quadtree nodes and leaves which has little contribution. Pruning is carried out in two ways: 1) SAQC is utilized like TC for CU skipping using dynamic thresholds; 2) For LCU with zero SAQC, tree nodes and leaves in depth2 and 3 are pruned; for 32x32 CUs with zero SAQC, tree leaves in depth 3 are pruned. Pruning is performed according to sufficient analysis.

Variable block size prediction makes motion representation more accurate and is adaptive to characteristics of input videos. It is known that stationary background and big moving objects are usually coded by large blocks. Regions with small moving objects tend to be further partitioned. During ME search, if a block can find a matching with little or no residual remained, further partition can be constrained. Reversely, if residual energy is high, it indicates further partition should be performed.

To measure whether ME matching for an LCU is good enough or not to disable further partition, sum of absolute quantized transform coefficient (SAQC) is used as a judging criterion. It is defined by

$$SAQC_{64x64} = \sum_{k=0}^{k=4} \left\{ \sum_{i=0}^{i=32} \sum_{j=0}^{j=32} | T \times Q(residual\_blk_{i,j}) | \right\}_k \tag{3.10}$$

where *residual_blk* is a 32x32 residual block and *T*x*Q* is integer DCT followed by quantization. HEVC adopts quadtree-based variable block size transform [12], however, only 32x32 DCT and quantization is used in Eq. (3.10) and SAQC$_{64x64}$ is defined as the sum of absolute coefficients of four quantized 32x32 residual blocks. Distributions of SAQC in all four depths are analyzed and shown in Figure 3.9. Similar to Figure 3.6 (a), the horizontal axes in Figure 3.9 represent number of LCUs in each depth. For either QP, upper limit of SAQC becomes higher as depth increases. Meanwhile, higher QP makes the upper limit of each depth in Figure 3.9 (b) smaller than that in (a).

Since SAQC ranges of different depths are only partially overlapped, similar to using

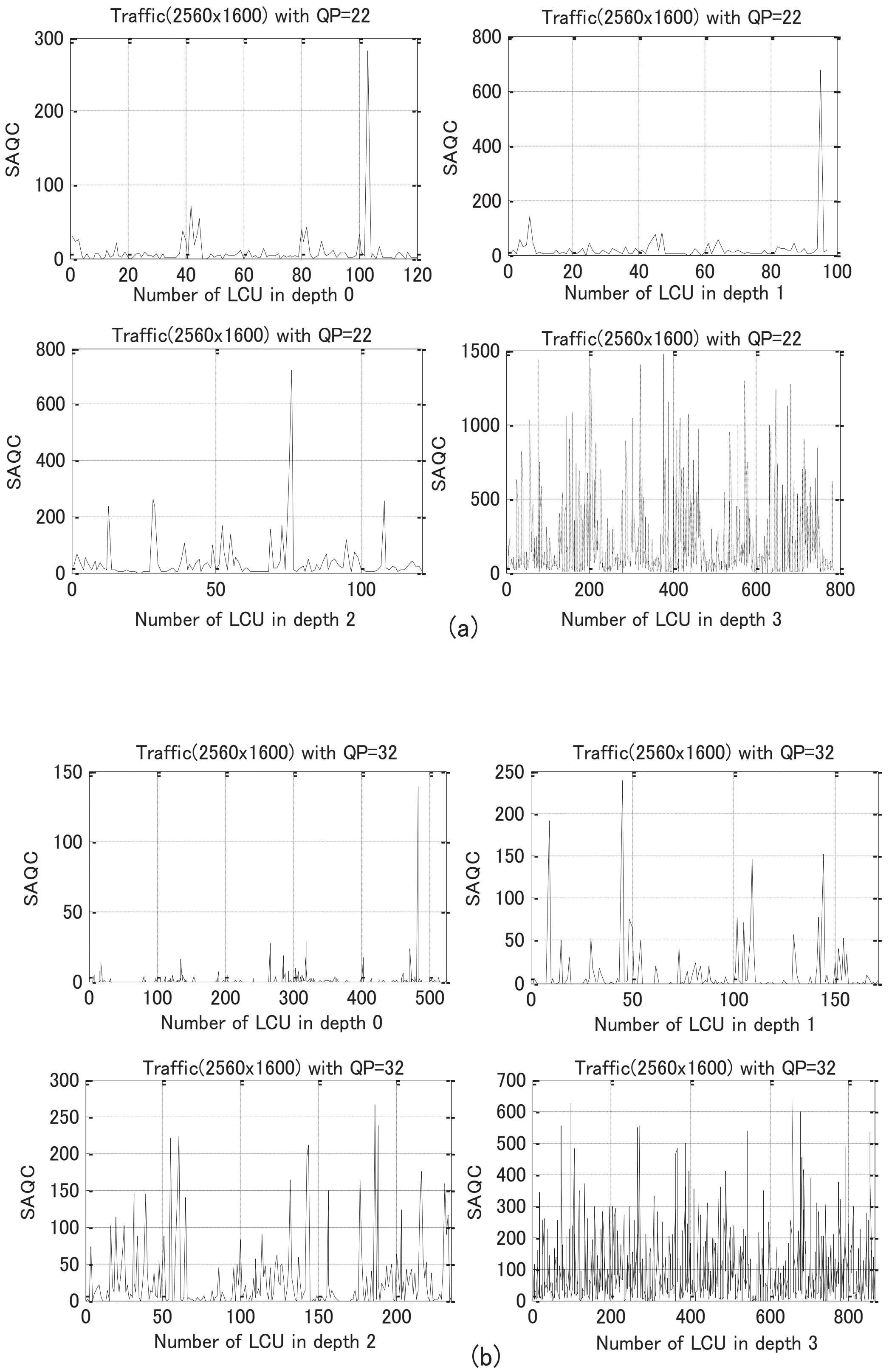


Figure 3.9 (a) SAQC distribution of four depths with QP=22.(b) SAQC distribution of four depths with QP=32

TC for skipping CUs, SAQC can be used to achieve the same purpose. When gathering TC values, SAQC values are also collected into four depths. The maximum SAQC of each depth is defined as upper limit (threshold) of this depth. Different from the calculation of TC range, lower limit is discarded and no adjustment is needed. Encoding flow using SAQC for depth skipping is identical to the lower half part of Figure 3.8, only that TC is substituted to SAQC. Another difference is that SAQC values are saved when encode each LCU in a frame. This is to update SAQC thresholds frame by frame.

It is common for a residual block quantized to all zero at low bit-rate coding. All-zero residual implies that both DC and AC signals are quantized to zero. This kind of block is called all-zero block (AZB). Figure 3.9 shows that all sub-figures contain a large amount of zero SAQCs, which indicates many AZBs exist. For an AZB, even part of further partitions are skipped, encoding performance can hardly be affected, since a little larger residual can be compensated by fewer bits used for signaling the choice of coding unit size. To verify this suggestion, simulation on how many 64x64 AZBs are coded by none depth-0 coding units is performed. Table 3.5 shows results on 4kx2k and 1080p sequences. The last four columns illustrate the percentage that actual depth of a 64x64 AZB is 0, 1, 2 and 3, respectively. It is noted that the majority of AZBs is coded by blocks in depth 0 and 1. Depth 2 and 3 only occupy a very small percentage, 2.39%

Table 3.5 Ratios of all-zero LCU coded to depth 0, 1, 2, 3, respectively.

Sequences	QP	Depth 0	Depth 1	Depth 2	Depth 3
crowdrun	22	0.0%	0.0%	0.0%	0.0%
crowdrun	37	80.08%	12.55%	4.76%	3.89%
into_tree	27	84.64%	12.2%	2.75%	0%
into_tree	37	97.42%	2.33%	0.24%	0%
sunflower	32	85.01%	10.89%	2.72%	1.36%
Sunflower	37	86.88%	8.16%	4.95%	3.18%
rushhour	22	68%	32%	0%	0%
rushhour	37	94.71%	3.96%	1.32%	0%
Average	**-**	**85.25%**	**11.78%**	**2.39%**	**0.57%**

and 0.57% respectively. Based on this analysis, coding blocks in depth 2 and 3 can be disabled for 64x64 AZBs.

For a non all-zero LCU, it is further partitioned into four 32x32 blocks. After ME, DCT and quantization, 32x32 AZBs may exist, which implies further partition might be constrained. To judge whether a quantized 32x32 block is AZB or not, criterion is defined as

$$SAQC_{32x32} = \sum_{i=0}^{i=32} \sum_{j=0}^{j=32} |T \times Q(residual\_blk_{i,j})| \tag{3.11}$$

where DCT and quantization size is 32x32 and other two transform sizes defined in Table 3.1 are not applied. Analysis about how many 32x32 AZBs are coded by depth-3 blocks is performed and less than 1.0% is observed in experiments. Hence, depth-3 coding blocks are skipped for 32x32 AZBs. In summary, zero SAQC based quadtree-pruning works as follows.

1) After 64x64 ME, 32x32 DCT & quantization, if an LCU is an AZB, depth-2 tree nodes and depth-3 leaves on the original quad-tree are pruned. Meanwhile, 16x16 DCT and quantization is discarded. Otherwise, the LCU is split into four 32x32 blocks and go to 2).

2) Perform 32x32 ME, DCT & quantization; if the SAQC of a 32x32 block is zero, all depth-3 leaves under corresponding depth-1 tree node are pruned. Meanwhile, 16x16 and 8x8 DCT & quantization are disabled. Otherwise, the 32x32 block is further partitioned.

3.3.4 Overall Processing Flow

The whole scheme consists of three parts which exploit temporal, spatial and transform domain potentials for acceleration, respectively. Figure 3.10 shows the diagram of an

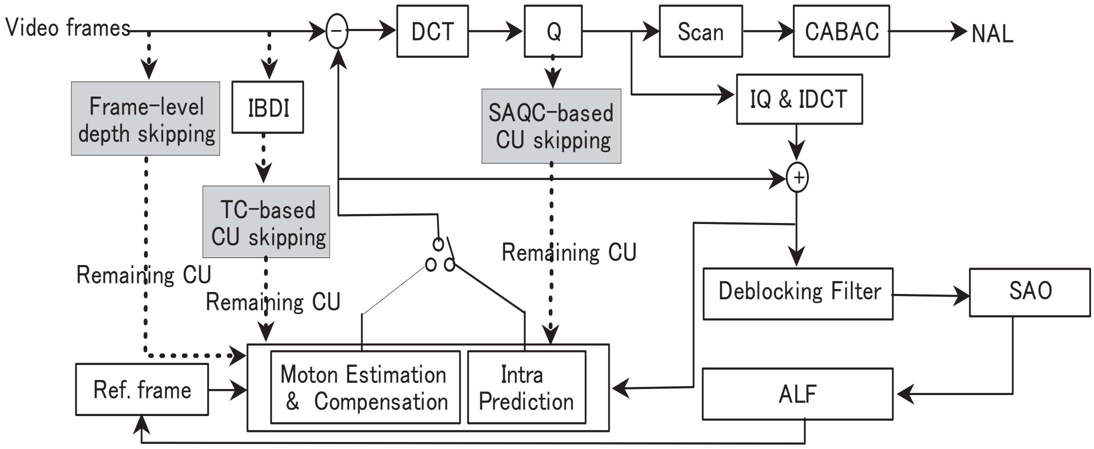


Figure 3.10 Proposed acceleration scheme comprised of three parts (shown in gray) in HEVC encoder.

HEVC encoder with proposed three methods shown in gray.

1) The "Frame-level depth skipping" takes advantage of temporal correlation to skip rarely adopted depths for all LCUs in consecutives frames.

2) The "TC-based CU skipping" measures spatial texture complexity (TC) of an LCU to filter out useless coding blocks. In this step, a dynamic threshold setting approach is proposed to make filtering adaptable to various videos.

3) The "SAQC-based CU skipping" fully utilizes sum of absolute quantized residual coefficient (SAQC) in transform domain to prune none-contributable quadtree nodes and leaves for both LCU and 32x32 CU.

For viewpoint of implementation, the whole algorithm can be divided into two parts: pre-optimization stage and optimization stage. Figure 3.11 shows the pre-optimization stage, which encodes each LCU in the first P frame as normal and gathers information like LCU's depth, TC and SAQC values. When encoding of the frame is completed, utilization ratio of each depth is computed; if rarely used coding depth exists, a flag signaling the depth is sent to the optimization stage. Also, using collected TC and SAQC values, dynamic thresholds are calculated by the method introduced in subsection 3.3.2 and 3.3.3. Optimization stage starts from the second P frame. Figure 3.12 shows the encoding flow for each LCU in this stage. Frame level depth skipping functions first followed by TC-based optimization. These two methods work before ME starts. If the number of skipped depths is no larger than 2 by above two methods, 64x64 ME is performed and optimization using SAQC is enabled. If no depth is skipped by using SAQC of an LCU, ME for its four sub-blocks are performed. Once an AZB is detected, the other transform blocks are disabled to avoid recursive residual encoding.

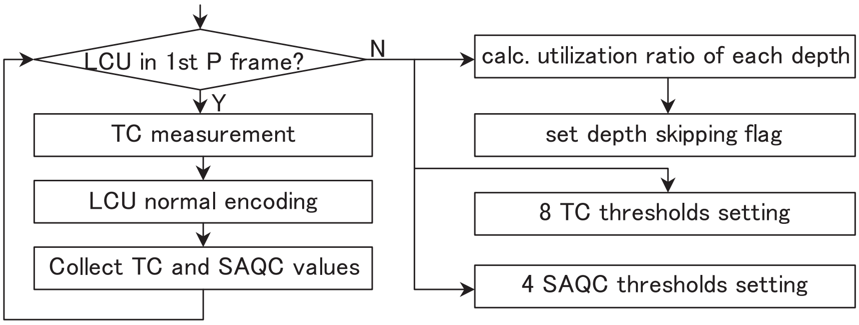


Figure 3.11 Pre-Optimization stage: calculation utilization ratio of each depth, TC thresholds setting and SAQC thresholds setting.

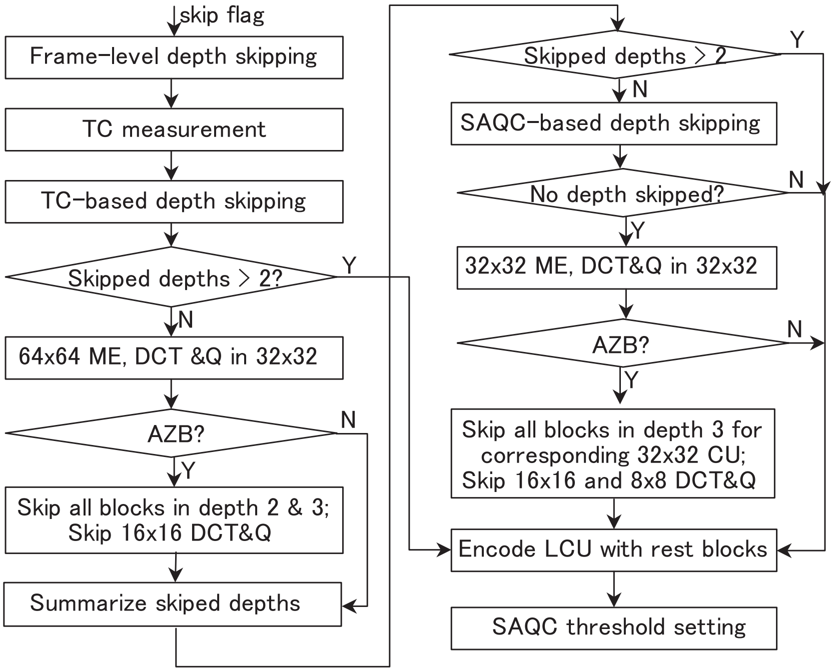


Figure 3.12 Optimization stage: optimization flow using proposed scheme.

For skipped luma block, processing for corresponding chroma blocks are also skipped. For skipped coding depth or block, all corresponding prediction blocks will be disabled for both intra and inter prediction process.

3.3.5 Performance Evaluation and Comparison Results

Proposed algorithm is implemented and verified in HM 4.0 [43]. Fourteen high resolution sequences with different motions and spatial features are tested using Low-Delay High-Efficiency profile [47]. Important coding parameters and tested sequences are listed in Table 3.6.

Table 3.6 Experimental parameters and tested video sequences.

Coding parameters	Test frame number=100; Max CU size = 64x64; MaxCUDepth=4; QP=22,27,32,37; GOP=IP..P, RQT depth Intra/Inter= 3; SR=64; InternalBitDepth=10; Entropy=CABAC; Fast Search=ON; Reference frame number=4 for 1080p, 2 for 4kx2k; RDOQ=ON; ALF=ON, FastEnc=ON;SAO=ON; AMP=OFF; NSQT=OFF; ECU=OFF
Video sequences	HD(1920x1080) : station2, sunflower, rushhour, parkscene, cactus, BasketballDrive, BQTerrace; 4kx2k(3840x2160): crowdrun, ducks_take_off, in_to_tree, old_town, parkjoy, peopleonstreet, traffic.

Comparison result of proposed algorithm against HM on seven 4kx2k sequences is shown in the right-hand column of Table 3.7. Among these sequences, peopleonstreet and traffic are cropped from 4kx2k video sources and are with resolution of 2560x1600 [47]. It is noted that the losses in terms of PSNR, BR and the gains in complexity reduction vary as QP changes. This is because our proposals perform differently at different QPs. For the same QP, encoding performances also differ for different sequences, since they differ in temporal motions and spatial textures. The highest encoding time reduction is almost always achieved at the lowest or highest QP for all sequences, as the second and third proposal tends to achieve higher coding reduction at low and high QP, respectively.

Table 3.7 Coding performance comparison with HM 4.0 and JCTVC-F092 [48] on 4kx2k sequences.

Sequences	Resolution	QP	JCTVC-F092[48]			Our proposal		
			ΔPSNR (dB)	ΔBR (%)	ΔTime (%)	Δ PSNR (dB)	Δ BR (%)	Δ Time (%)
crowd_run	3840x2160	22	-0.001	0.14	-0.93	-0.008	0.276	-26.463
		27	-0.006	0.05	-2.17	-0.005	0.244	-27.390
		32	-0.011	-0.09	-8.01	-0.003	0.373	-21.731
		37	-0.017	-0.32	-38.79	-0.005	0.761	-23.941
Average			**-0.008**	**-0.055**	**-12.475**	**-0.005**	**0.413**	**-24.881**
ducks_take_off	3840x2160	22	0.000	0.01	-0.27	0.000	1.035	-42.552
		27	0.001	0.06	-1.18	-0.007	1.105	-22.178
		32	0.000	-0.09	-3.31	-0.012	0.248	-30.320
		37	-0.001	-0.16	-9.47	-0.004	-0.089	-32.637
Average			**0.000**	**-0.045**	**-3.557**	**-0.005**	**0.574**	**-31.922**
In_to_tree	3840x2160	22	-0.002	0.198	-2.78	-0.012	0.101	-21.196
		27	-0.005	0.332	-6.84	-0.016	-0.640	-23.311
		32	-0.008	-0.290	-45.26	0.000	0.000	-33.257
		37	-0.013	-1.368	-64.56	-0.004	-0.126	-61.896
Average			**-0.007**	**-0.282**	**-29.860**	**-0.008**	**-0.166**	**-34.915**
old_town_cross	3840x2160	22	0.001	0.09	-0.10	-0.005	0.684	-44.358
		27	-0.003	-0.11	-6.14	-0.001	-0.206	-40.177
		32	-0.007	-1.36	-45.50	0.000	0.036	-30.096
		37	-0.009	-1.40	-63.18	-0.005	0.000	-56.911
Average			**-0.005**	**-0.695**	**-28.730**	**-0.002**	**0.128**	**-42.886**
park_joy	3840x2160	22	0.006	-0.44	-4.45	-0.008	0.777	-34.697
		27	0.008	-0.38	-9.95	-0.017	1.465	-30.801
		32	0.007	-0.45	-15.47	-0.001	-0.012	-20.906
		37	0.005	-0.49	-23.26	0.000	-0.023	-26.778
Average			**0.006**	**-0.440**	**-13.282**	**-0.006**	**0.551**	**-28.296**
Peopleonstreet	2560x1600	22	-0.006	-0.13	-0.58	-0.048	0.147	-34.024
		27	-0.011	-0.24	-3.32	-0.056	1.413	-37.562
		32	-0.013	-0.34	-8.48	-0.035	0.478	-36.141
		37	-0.015	-0.35	-14.64	-0.041	0.768	-40.883
Average			**-0.011**	**-0.265**	**-6.755**	**-0.045**	**0.701**	**-37.153**
Traffic	2560x1600	22	-0.016	-0.52	-8.04	-0.024	1.672	-35.014
		27	-0.026	-0.16	-20.85	-0.006	0.194	-25.176
		32	-0.037	-1.53	-38.16	-0.034	-0.139	-35.947
		37	-0.032	-1.31	-50.46	-0.024	0.293	-45.962
Average			**-0.028**	**-0.880**	**-29.377**	**-0.022**	**0.505**	**-35.525**
Total average			**-0.008**	**-0.380**	**-17.719**	**-0.013**	**0.386**	**-33.654**

The average time reduction in Table 3.7 is with least gain of 24.8% for *crowdrun* and most gain of 42.8% for *oldtown*, respectively. *crowdrun* is a sequence with hundreds of running people in the foreground and many tree leaves and small running people in the background. Generally, this sequence is difficult for compression. As to the performance of video quality, PSNR loss by proposed scheme is always less than 0.06 dB in all the cases. Compared with original HEVC, proposed scheme is able to achieve 33.65% encoding time reduction with an average of 0.38% bit increment and PSNR drop is 0.013 dB.

Table 3.7 also compares our proposal and ECU technique in JCTVC-F092 [48]. Both our proposal and [48] are evaluated against original HM 4.0 by the parameters of △PSNR, △Bit-rate and that defined in Eq. (3.13). As observed, [48] can achieve very high complexity reduction for sequences with smooth texture or simple motion when QP is 32 and 37. However, for sequences with complex texture, motion and for low QP cases, complexity reduction from ECU is deteriorated. This is due to the reason that ECU only takes advantage of SKIP mode while SKIP mode happens infrequently for complex texture, motion and for low QP. For *ducks_take_off* and peopleonstreet (the former contains complex texture, the later contains complex motion), [48] only reduces 3.557% and 6.755% of encoding time for the two sequences. Nevertheless, our proposal accelerates the encoder by 31.922% and 37.153%, respectively. Besides, for sequences with smooth texture such as old_town_cross and in_to_tree encoded with QP equal to 22, time reduction by [48] is also limited, only 0.10% and 2.78% for the two sequences respectively, because SKIP mode is rarely selected. On the contrary, our proposal utilizes multiple features in spatial, temporal and transform domains and it can reduce a large proportion of complexity for sequences with a wide range of content characteristics. Averagely, our proposal reduces 15.8% more complexity than [48] does. As to performances of bit-rate, [48] saves about 0.38% bit rate for HM 4.0 and our proposal increases about the same amount. The increment only requires slightly more bits to be transferred.

Table 3.8 shows comparisons between our proposal and a similar work [39] when QP is 24. The simulation is carried out on seven 1920x1080 video sequences. Among them, BasketballDrive and parkscene are with fast motion; cactus is a sequence with irregular motions; other sequences are with middle-level or slow motion and homogenous background. For sequences with fast motions, work [39] causes large bit-rate increment, 2.9% and 1.9% for BasketballDrive and parkscene, respectively. On the contrary, bit-rate increment of our proposal is less than 0.9%. For sequences with slow and homogenous motions like statoin2 and BQTerrace, our proposal gains much higher reduction than [39] does, too. Complexity reduction in [39] is about 18.6% to 27.41% for all sequence, since only one coding depth is skipped by the method in [39]. In all, proposed scheme outperforms [39] in all cases and achieves a further reduction of 9.3% averagely.

Table 3.8 Coding performance comparison with [39] on 1080p videos.

Sequences	Reference [39]			Proposed work		
	ΔPSNR (dB)	ΔBR (%)	ΔTime (%)	ΔPSNR (dB)	ΔBR (%)	ΔTime (%)
staion2	-0.01	0.00	-25.15	-0.008	0.66	-44.29
sunflower	-0.02	0.10	-20.22	0.001	0.15	-31.17
rushhour	0.01	0.10	-27.41	-0.005	0.24	-29.29
parkscene	-0.02	1.90	-20.43	-0.012	0.56	-24.72
cactus	-0.02	1.63	-20.62	-0.013	0.30	-26.07
BQTerrace	-0.02	1.15	-28.86	-0.023	0.48	-48.28
BasketballDrive	-0.02	2.93	-18.66	-0.012	0.86	-25.73
Average	**-0.014**	**1.11**	**-23.05**	**-0.010**	**0.46**	**-32.36**

Table 3.9 shows the effectiveness of each proposal in different coding scenario. Here, effectiveness is calculated as number of skipped coding depths. For a test sequence, skipped CUs in an LCU by our proposal are usually with variable number and variable size. To calculate precisely how many depths are skipped by each proposal, effectiveness of skipping a certain CU must be defined and it is specified as follows. If a 64x64 CU is skipped, effectiveness is 1; if all four 32x32 CUs in an LCU are skipped, effectiveness is also 1; in other words, skipping one 32x32 CU gives 1/4; if all the 16

Table 3.9 Effectiveness of each proposal in different coding scenario.

Sequence	QP	Frame-level	TC-based	SAQC-based
BQTerrace	22	0.00	2.04	0.15
	27	0.00	1.27	0.29
	32	0.00	1.03	0.35
	37	0.00	1.12	0.68
Average	**-**	**0.00**	**1.36**	**0.37**
Station2	22	0.00	1.34	0.11
	27	1.00	0.47	0.35
	32	1.00	0.50	0.46
	37	1.00	0.42	0.63
Average	**-**	**0.75**	**0.68**	**0.39**
in_to_tree	22	0.00	0.65	0.28
	27	0.00	0.54	0.30
	32	1.00	0.26	0.37
	37	2.00	0.32	0.46
Average	**-**	**0.75**	**0.44**	**0.35**
Total Average	**-**	**0.50**	**0.83**	**0.37**

16x16 CUs in an LCU are skipped, effectiveness is also counted as 1, which means effectiveness is 1/16 for skipping one 16x16 CU; similarly, the effectiveness of skipping one 8x8 CU is 1/64. Thereafter, effectiveness is defined as

$$Effectiveness = \frac{64\times \mathrm{a} + 16\times \mathrm{b} + 4\times \mathrm{c} + \mathrm{d}}{64\times Num\_of\_LCU} \tag{3.12}$$

where a, b, c and d are skipped numbers of 64x64, 32x32, 16x16 and 8x8 CUs, respectively. *Num_of_LCU* is the total number of LCUs in a test sequence. *Effectiveness* denotes skipped depths per LCU. Since 4 depths can be skipped at maximum, *Effectiveness* is a number in the range of [0, 4].

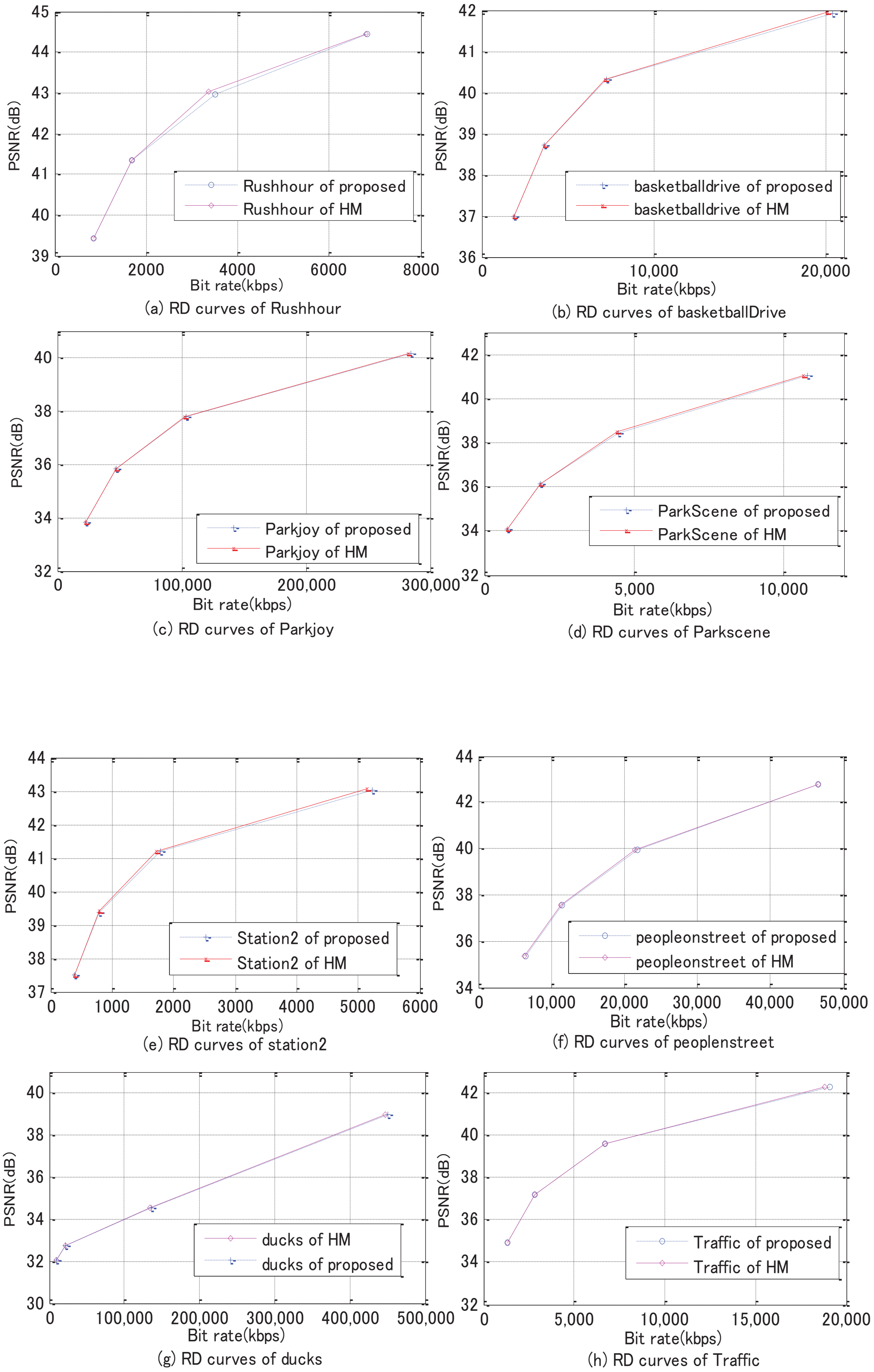
As disclosed in Table 3.9, frame-level method can skip a whole depth for staion2 and in_to_tree when QP is large; 2 depths are completely skipped for in_to_tree when QP is 37. TC-based method is most effective when QP is 22, as TC ranges are less affected by quantization which facilitates more CUs to be skipped. SAQC-based proposal is capable of pruning unnecessary CUs especially for large QP values. Another phenomenon disclosed is that part of LCUs can be optimized by multiple proposals. As a conclusion, all proposals are indispensable for complexity reduction for videos with different characteristics in texture and motion.

Encoding performances represented by RD curve on eight representative sequences are shown in Figure 3.13. The bit-rate increment of these selected sequences is the largest among all tested video sequences. The largest bit-rate increment is observed for rushhour at QP equal to 32. For all other seven sequences, the RD curves from our proposed scheme and that from HEVC are just closely overlapped. To sum up, for sequences with different texture complexity and motion features, proposed algorithm achieves almost always the same performance compared with original HEVC encoder.

3.4 Conclusion

This chapter firstly brings forward a two-stage PU size decision algorithm to speed up the original intra coding in HEVC. In the pre-stage, texture complexity of down-sampled largest coding unit (LCU) and its four sub-blocks are analyzed according to video content, to filter out unnecessary prediction units for both the LCU and its sub-blocks. Secondly, during intra coding, prediction unit sizes of encoded neighboring blocks are utilized to skip small prediction unit candidates for current block. Experimental results show that proposed algorithm can speed up original HEVC intra coding by a factor of up to 73.7% and by averagely 44.91% for 4kx2k video sequences. Meanwhile, the peak signal-to-noise ratio degradation is less than 0.04dB and bit-rate stays almost the same compared with that of original HEVC intra coding.

Next, this chapter proposes a scheme to accelerate the quadtree-based variable block size prediction for the emerging video coding standard of HEVC. It consists of three parts which exploit temporal, spatial and transform-domain potentials for acceleration



(a) RD curves of Rushhour

(b) RD curves of basketballDrive

(c) RD curves of Parkjoy

(d) RD curves of Parkscene

(e) RD curves of station2

(f) RD curves of peoplenstreet

(g) RD curves of ducks

(h) RD curves of Traffic

Figure 3.13 Comparisons of RD curves

respectively. The “frame-level depth skipping” method which exploits temporal correlation is an ultra fast way to skip rarely used coding depth for consecutive frames. The “TC-based skipping” method takes advantage of texture complexity in spatial domain to filter out unsuitable coding blocks at LCU-level. A novel threshold setting technique is also proposed to make pruning adaptable to video signals. The “SAQC-based skipping” makes use of transform-domain feature to cut tree nodes & leaves for LCU and 32x32 CU. The above three methods form an optimizing scheme which can accelerate various videos containing all kinds of features in textures and motions. Experimental results demonstrate that proposed algorithm can speed up original HEVC prediction process by a factor of up to 61.9% for 4kx2k sequences. Meanwhile, coding quality degradation and bit-rate increment are very trivial. This work also outperforms existing work with more complexity reduction and less bit-rate increment. As our acceleration is achieved by exploiting temporal, spatial and transform-domain features in video signals, the proposed scheme can also be useful for reducing the computational complexity for low-delay B configuration.

4. All-ZERO BLOCK-BASED OPTIMIZATION FOR QUADTREE-STRUCTURED PREDICTION AND RESIDUAL ENCODING

4.1 Background and Previous Proposals

After quantization, if all coefficients of a residual block are zero, the block is called an all-zero block (AZB). In the literature on H.264/AVC, AZB detection algorithms are designed such that discrete cosine transform (DCT) and quantization can be skipped. Due to this reason, lots of works tried to detect AZB for the purpose of reduce computational complexity of a video encoder.

4.1.1 Quadtree-based Prediction and Residual Transform

The new framework consisting of CU, PU and TU for image partition, motion compensation and residual encoding respectively, are illustrated by Figure 4.1. The left square is a 64x64 LCU partitioned into CU with sizes from 8x8 to 32x32. For each 2Nx2N CU, PU sizes can be 2Nx2N, 2NxN and Nx2N; RQT for each PU has three

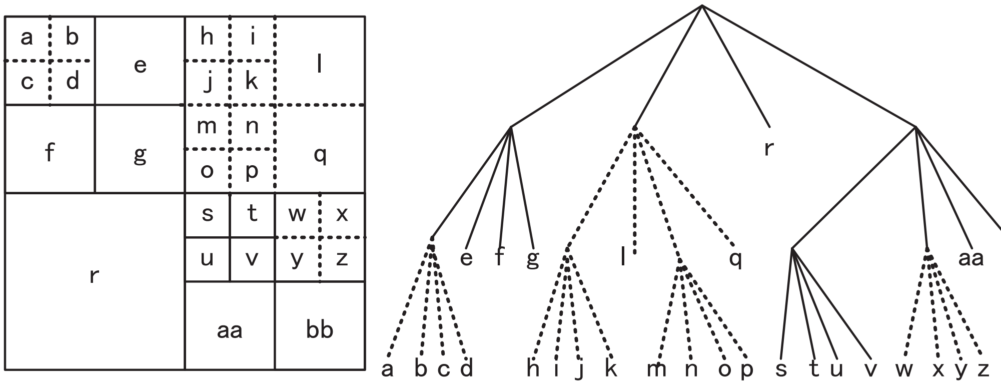


Figure 4.1 Quadtree-based LCU splitting and residual Quadtree transform.

levels from 2Nx2N to N/2xN/2. In Figure 4.1, recursive splitting of LCU into CU is depicted in solid lines and each CU is predicted in depth-first alphabetical order. TUs and their corresponding RQT are denoted by dashed lines.

Quadtree-based Prediction mainly is responsible to find the optimal combination of prediction block sizes for image areas comprised of unpredictable textures and motions. And the prediction blocks is called matching block for original image block. Then, the difference between original block and its predicted matching block will be generated, DCT transformed. Next, the transformed coefficients will be entropy coded and compressed into bit stream for storage or transmission.

The integer DCT transform in HEVC draft 6 is an approximation of DCT specified as a matrix multiplication in two stages: horizontal and vertical. Eq. (4.1) shows the general form of an N×N block transformed by DCT.

$$Y_{N\times N} = (D\,(X_{N\times N} D^T \otimes PF_1)) \otimes PF_2,\ N = 4, 8, 16, 32 \tag{4.1}$$

where Y is the transformed result, D is a near-orthogonal DCT core, X is a residual block, and all blocks are of size N×N. $\otimes$ stands for scalar multiplication. $PF_1$and PF_2 are scaling operations defined in Eqs. (4.2) and (4.3), respectively.

$$PF_1 : y = (x + (1 << (M - 2 + B - 8))) >> (M - 1 + (B - 8)) \tag{4.2}$$

$$PF_2 : y = (x + (1 << (M + 5))) >> (M + 6) \tag{4.3}$$

where $M = log_2(N)$, and B is the internal bit depth of the video signal. The scaling operations ensure a bit width of 16 after each transform stage. All computations in Eq. (4.1) can be implemented using only addition and shift operations.

The 4×4 integer transform is specified as Eq. (4.4). Note that the 4×4 transform core is completely different from that in H.264/AVC. Transform cores for other sizes are aslo

$$Y=\begin{bmatrix}64 & 64 & 64 & 64\\ 83 & 36 & -36 & -83\\ 64 & -64 & -64 & 64\\ 36 & -83 & 83 & -36\end{bmatrix}\left(\begin{bmatrix}X\end{bmatrix}\begin{bmatrix}64 & 83 & 64 & 36\\ 64 & 36 & -64 & -83\\ 64 & -36 & -64 & 83\\ 64 & -83 & 64 & -36\end{bmatrix}\otimes PF_1\right)\otimes PF_2 \tag{4.4}$$

re-designed [53]. After DCT, the transformed coefficient is quantized using operations in Eq. (4.5).

$$Q=sign(Y)((|Y|\times f(QP\,\%\,6)+F<<(qbs-9))>>qbs \tag{4.5}$$

$$qbs=21+\left\lfloor\frac{QP}{6}\right\rfloor-\log_2 N \tag{4.6}$$

where Q is the quantized result and QP is a quantization parameter ranging from 0 to 51. F is an offset used for quantization, which is 171 for intra coding and 85 for inter prediction. The parameter qbs, defined in Eq. (4.6), is the number of bits for right shift. The symbol $\lfloor\,\rfloor$ denotes rounding to the nearest integer. The function f is a scalar multiplier specified in Eq. (4.7), with x ranging from 0 to 5.

$$f(x)=\{26214,23302,20560,18396,16384,14564\} \tag{4.7}$$

4.1.2 Previous Proposals for All-Zero Blocks Detection

Sousa [49] derives an 8x8 AZB detection condition by comparing sum of absolute difference (SAD) to the theoretical upper limit of transformed coefficients in conventional DCT. Formally, the upper limit is calculated as the right hand-term of Eq. (4.8).

$$\begin{aligned}|F(u,v)|&=K(u)K(v)\times\left|\sum_{x=0}^{7}\sum_{y=0}^{7}f(x,y)\right|\times\cos(\frac{(2x+1)u\pi}{16})\times\cos(\frac{(2y+1)v\pi}{16})\\ &\le\frac{1}{4}\times\cos^2(\frac{\pi}{16})\times SAD_{8\times 8}\end{aligned} \tag{4.8}$$

Eq.(4.8) can be easily re-written as

$$\left\lfloor \frac{|F(u,v)|}{\Delta} \right\rfloor < 1 \Leftrightarrow SAD_{8\times 8} < 4\times\Delta\times\cos^{-2}(\frac{\pi}{16});\ \ \Delta \text{ is } \textit{quantization scale} \tag{4.9}$$

By comparing SAD with the limit, if the SAD of an 8x8 block is less than right-hand term of Eq. (4.9), the block will be definitely transformed to all zero by DCT. As traditional floating-point DCT is used for calculating the upper limit, there will be no false detection. Another advantage of [49] is that its detection condition can be applied to any DCT-based codec, H.261, H.263, H.264/AVC as long as HEVC. However, one problem is that the detection condition in Eq. (4.9) is too strict which leads to miss detections.

Works [50]-[52] developed other AZB detection conditions dedicated for H.264/AVC. In H.264/AVC, due to the simple 4x4 and 8x8 DCT matrix with element equal to 1 or 2, Moon et al. [50] derived individual conditions for each frequency component to detect 4×4 AZBs. Lee and Lin [51] improved detection accuracy by using a near-sufficient condition. Liu et al. [52] proposed the detection of 4×4 AZBs using partial SAD. One common feature among [50]-[52] is that they strongly rely on the 4×4 or 8×8 DCT cores and quantization rule in H.264/AVC. As DCT cores and quantization rule were re-designed in HEVC [53], the methods described in [50]-[52] became unsuitable for detecting AZBs for the draft 6 of HEVC standard.

4.1.3 Previous Proposals for Quadtree-based Prediction Optimization

To speed up the quadtree-structured prediction in HEVC, [39] compares utilization ratios of two continuous depths in frame level and CU level, and skips the depth much less utilized. With fast motions and complicated scenes, however, the complexity saving of [39] deteriorate. Work [54] defines a feature vector comprised of variance and SATD of residual as well as RD cost. Using the vector, conditional probabilities of split and non-split are calculated and the one with larger probability is chosen for encoding. As the training process cannot cover all possible cases, performance of [54] will be affected when features of input videos are different from those of training samples.

4.2 Proposed All-Zero Block Detection Algorithms

As explained in section 4.1.2, AZB detection mainly predicts a block by taking advantage of SAD. In section 4.2.2, a SAD-based algorithm is firstly developed to detect AZB. The proposed algorithm is able to achieve much better detection rate than [49] can. Then, another AZB detection algorithm is developed in section 4.2.3 by using the theory of energy conversation.

4.2.1 Motivations of using AZB

From previous works [50]-[52], it is known that detection of AZB can skip unnecessary DCT and quantization so that encoding complexity can be reduced. However, according to our intensive experiments, 80% of encoding complexity of HEVC is taken by hybrid prediction. That is to say, skipping DCT and quantization can only achieve less than 20% reduction.

Except for skipping unnecessary DCT and quantization, AZB should be used in other ways. It is natural that if a block can find its predicted one without introducing any residual, the block must an AZB. In this case, there is no need to further partition the block and thus motion estimation and compensation for sub-block can be saved. In addition, during motion estimation, if SAD approaches the detection condition of AZB, further searching can be early terminated.

In natural videos, there are always homogenous areas and continuous frames with stationary motions. In these cases, AZB can be utilized for computational complexity optimization. Of course, for high QP coding, AZB blocks always occupies a very large amount.

4.2.2 Sum of Absolute Difference based AZB Detection (M1)

According to Eq. (4.5), if a transformed coefficient is quantized to zero, the following condition must be satisfied.

$$|Q| < 1 \Leftrightarrow |Y| < \frac{2^{(qbs-9)} \times (2^9 - F)}{f(QP\,\%\,6)} \tag{4.10}$$

To avoid division operation, a new function *g* is defined as

$$g(x) = \{40,45,51,57,64,72\},\ x = 0 \sim 5 \tag{4.11}$$

For the same *x*,

$$f(x) \times g(x) \approx 2^{20} \tag{4.12}$$

Accordingly, Eq. (4.10) can be rewritten as

$$|Y| < 2^{(qbs-29)} \times (2^9 - F) \times g(QP\,\%\,6) \tag{4.13}$$

Substituting *qbs* with the right-hand term in Eq. (4.6), the condition for AZB detection can be derived as

$$|Y| < ((2^9 - F) \times g(QP\,\%\,6)) >> (\log_2 N + 8 - \left\lfloor \frac{QP}{6} \right\rfloor) \tag{4.14}$$

Demonstrably, if all the coefficients of a transformed block satisfy Eq. (4.14), the block is an AZB. However, checking all coefficients will cause nonnegligible computational overhead. Therefore, the SAD of a residual block is checked instead. We illustrate this process using the 4×4 transform in Eq. (4.4). Each transformed coefficient Y_{mn} ($0 \le m,n \le 3$) in a 4×4 block Y is a linear combination of all residual signals:

$Y_{mn} = \sum_{i=0}^{3}\sum_{j=0}^{3}\left(C_{ij}^{mn} \cdot X_{ij}\right)$, C_{ij}^{mn} is the coefficient of X_{ij} in spatial position (i, j) in Y_{mn}. The summation of all DCT transformed coefficients is also a linear combination of all residual signals, expressed as

$$
\begin{aligned}
\sum_{m=0}^{3}\sum_{n=0}^{3} Y_{mn} &= \sum_{m=0}^{3}\sum_{n=0}^{3}\sum_{i=0}^{3}\sum_{j=0}^{3}\left(\mathrm{C}_{ij}^{mn} \cdot X_{ij}\right) = \sum_{i=0}^{3}\sum_{j=0}^{3}\left(\sum_{m=0}^{3}\sum_{n=0}^{3}\mathrm{C}_{ij}^{mn}\right)\cdot X_{ij} \\
&= \sum_{i=0}^{3}\sum_{j=0}^{3} S_{ij}\cdot X_{ij} = S_{00}\cdot X_{00} + S_{01}\cdot X_{01} + \cdots + S_{32}\cdot X_{32} + S_{33}\cdot X_{33}
\end{aligned}
\tag{4.15}
$$

where $S_{ij} = \sum_{0\le m,n\le 3} C_{ij}^{mn}$, which is the sum of coefficient of X_{ij} in all transformed coefficients. The maximum value of S_{ij} is defined as

$$
S_{\max} = \max\{\ |S_{00}|,\ |S_{01}|,\ |S_{02}|, \cdots |S_{31}|,\ |S_{32}|,\ |S_{33}|\ \}
\tag{4.16}
$$

Next, we define the absolute average of all Y_{mn} as Y_{av}, and the upper limit of Y_{av} is derived as

$$
\begin{aligned}
|Y_{av}| &= \frac{1}{16}\cdot\left|\sum_{m=0}^{3}\sum_{n=0}^{3} Y_{mn}\right| \le \frac{1}{16}\cdot\sum_{i=0}^{3}\sum_{j=0}^{3}|S_{ij}|\cdot|X_{ij}| \\
&\le \frac{S_{\max}}{16}\cdot\sum_{i=0}^{3}\sum_{j=0}^{3}|X_{ij}| = \frac{S_{\max}}{16}\cdot SAD_{4\times4}
\end{aligned}
\tag{4.17}
$$

If a 4×4 residual block is an AZB, Y_{av} must satisfy the inequality in Eq. (4.14), which gives

$$
|Y_{av}| < ((2^9 - F)\times g(QP\,\%\,6)) >> (\log_2 N + 8 - \left\lfloor \frac{QP}{6} \right\rfloor)
\tag{4.18}
$$

Substituting the left-hand term of Eq. (4.18) with the last term of Eq. (4.17), the 4×4 AZB detection condition is derived as

$$
SAD_{4\times4} < \frac{16\times((2^9 - F)}{S_{\max}} \times g(QP\,\%\,6)) >> (\log_2 N + 8 - \left\lfloor \frac{QP}{6} \right\rfloor)
\tag{4.19}
$$

Since 4×4 to 32×32 transform cores are given in [53], for any N×N residual block, S_{max}

is a fixed value calculated by the matrix multiplication defined in Eq. (4.1). The final AZB detection conditions for variable block sizes are derived as

$$\begin{cases} SAD_{4\times4} < \dfrac{2^{B+5}\times(2^9-F)}{61009}\times g(QP\%6) >> (10-\left\lfloor\dfrac{QP}{6}\right\rfloor) \\ SAD_{8\times8} < \dfrac{2^{B+9}\times(2^9-F)}{228741}\times g(QP\%6) >> (11-\left\lfloor\dfrac{QP}{6}\right\rfloor) \\ SAD_{16\times16} < \dfrac{2^{B+9}\times(2^9-F)}{55225}\times g(QP\%6) >> (12-\left\lfloor\dfrac{QP}{6}\right\rfloor) \\ SAD_{32\times32} < \dfrac{2^{B+15}\times(2^9-F)}{866761}\times g(QP\%6) >> (13-\left\lfloor\dfrac{QP}{6}\right\rfloor) \end{cases} \tag{4.20}$$

This section derives a mathematical condition for AZB detection, based on the DCT cores and quantization rule in HM 6.0. The derived condition can be used to detect AZBs in variable sizes ranging from 4×4 to 32×32 with high detection accuracies.

4.2.3 Energy Conservation Based Detection (M2)

Since the energy of signals before and after DCT is conserved, the following equality is satisfied.

$$\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\left|Y(i,j)\right|^2 = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\left|x(i,j)\right|^2 \tag{4.21}$$

where x is a residual block and Y stands for the transformed result by traditional floating-point DCT. The transform cores of 4x4 to 32x32 for integer DCT in HEVC are redesigned and they satisfy

$$D_{N\times N}\times D_{N\times N}^{T} = I_{N\times N}\times 2^{12+\log_2 N} \tag{4.22}$$

Where I is an NxN unit matrix, D is a DCT core and T means matrix transpose. Take the two post-scaling operations defined in Eq.(4.2) and (4.3) into consideration, the energy before and after integer DCT in HEVC should satisfy the following condition.

$$D_{N\times N}\times D_{N\times N}^{T}\otimes PF_1\otimes PF_2=I\times 2^{7-\log_2 N} \tag{4.23}$$

Then, the energy conservation equation is derived as

$$\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\left|Y(i,j)\right|^2=\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\left|x(i,j)\right|^2\times 2^{2\times(7-\log_2 N)} \tag{4.24}$$

If a residual block is an AZB, the sufficient condition is that each coefficient satisfies the condition in Eq.(4.14), thus the following two conditions should be satisfied.

$$\frac{1}{N}\times\left[\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}x(i,j)\right]\times 2^{7-\log_2 N}\ <\ T_{dc} \tag{4.25}$$

$$\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}x(i,j)^2-\left|\frac{1}{N}\times\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}x(i,j)\right|^2\ <\ T_{ac},\quad T_{ac}=(k\ll(14-2\times\log_2 N))\times {T_{dc}}^2 \tag{4.26}$$

where, T_{dc} is the right hand term of Eq.(4.25), k is an energy factor which depends on the size of transform cores. Through intensive simulations, k is set to 2, 4, 5 and 9 for 4x4, 8x8, 16x16 and 32x32 transforms, respectively. Using Eq.(4.25) and Eq.(4.26), AZB can be detected if residuals can satisfy both equations.

4.2.4 Detection Accuracy of Proposed Methods

To evaluate the efficiency of an AZB detection method, the detection accuracy (DA) for N×N AZB is defined by

$$DA_{N\times N}=\frac{Correctly Detected\ No.\ of\ N\times N\ AZBs}{Total\ No.\ of\ N\times N\ AZBs}\times 100\% \tag{4.27}$$

The larger $DA_{N\times N}$ is, the more AZBs are detected. Table 4.1 lists comparisons of DA performance of M1 with a similar work [49]. Compared with other existing works, AZB detection method in [49] is independent from specific DCT matrix and can be extended to detect AZB in variable sizes. Thus, we choose it as anchor for comparison. In Table

Table 4.1 *DA* performance comparison of M1 with Sousa's method [49].

Sequences	QP	Sousa's method [49]						Proposed AZB detection method					
		Luma Y			Chroma U/V average			Luma Y			Chroma U/V average		
		8×8	16×16	32×32	4×4	8×8	16×16	8×8	16×16	32×32	4×4	8×8	16×16
flowervase (416×240)	22	59.59	34.87	34.07	95.84	87.83	80.18	87.18	68.76	44.69	99.61	96.19	90.96
	27	64.66	38.81	27.93	97.04	89.65	80.29	89.62	75.81	43.54	99.86	97.97	92.91
	32	62.04	33.53	23.33	97.48	89.04	80.89	93.21	**79.69**	53.33	100.0	99.28	97.43
	37	74.03	49.36	23.11	99.46	95.63	88.61	96.55	83.26	**64.84**	100.0	99.99	98.42
Basketball drill (832×480)	22	0.24	0.00	0.00	68.92	8.01	0.00	12.14	0.02	0.00	96.44	**71.73**	5.96
	27	2.36	0.00	0.00	79.32	33.39	0.95	**50.04**	2.19	0.00	98.74	**87.74**	**41.96**
	32	2.08	0.00	0.00	84.00	44.41	0.82	**81.45**	8.71	0.00	99.50	**97.63**	**78.61**
	37	17.59	0.35	0.00	96.44	74.93	31.05	**92.83**	**47.27**	0.38	99.34	98.59	**89.97**
Vidyo4 (720p)	22	16.01	5.35	6.19	91.85	47.00	1.99	**71.71**	18.30	7.52	99.86	**98.01**	**55.80**
	27	34.90	6.57	3.78	95.84	70.36	23.02	**83.17**	**47.03**	7.87	99.97	99.85	**88.99**
	32	39.76	6.76	3.27	98.02	77.96	27.41	**95.46**	**62.82**	27.63	99.98	99.83	**99.23**
	37	61.56	26.91	3.90	99.93	96.30	68.05	98.77	**83.09**	39.97	99.99	99.81	99.13
Kimono (1080p)	22	0.68	0.23	0.00	63.92	7.76	0.79	18.31	1.26	0.50	97.41	**61.97**	9.84
	27	6.11	0.27	0.04	80.13	29.13	2.61	**80.21**	9.16	0.39	99.18	**94.93**	36.15
	32	6.67	0.18	0.02	90.71	36.88	2.56	**93.91**	**45.10**	3.53	98.02	**98.60**	**86.95**
	37	42.00	3.48	0.11	99.03	85.66	23.97	**97.30**	**77.62**	17.70	99.08	98.62	**97.32**
Cactus (1080p)	22	0.00	0.00	0.00	45.86	1.45	0.00	0.12	0.00	0.00	**87.03**	41.08	9.05
	27	0.03	0.00	0.00	67.49	18.25	0.05	38.26	0.00	0.00	98.14	**73.92**	33.59
	32	0.04	0.00	0.00	79.48	25.62	0.11	**75.59**	15.14	0.00	99.59	**98.00**	**84.50**
	37	29.23	0.00	0.00	97.80	66.72	17.23	**91.32**	**54.83**	1.91	99.70	100.0	**90.58**
Traffic (2560×1600)	22	1.06	0.07	0.00	57.10	8.07	0.00	**46.05**	0.57	0.00	92.93	**54.15**	8.22
	27	13.32	0.09	0.00	73.92	17.87	2.48	**70.75**	12.74	0.00	98.39	**66.49**	30.28
	32	12.50	0.02	0.00	82.55	33.38	3.07	**87.45**	36.83	1.26	99.76	**97.51**	**66.50**
	37	36.65	3.34	0.00	97.84	71.00	22.94	**95.96**	**58.24**	13.11	99.89	100.0	**95.09**
Average	-	24.29	8.75	5.23	84.99	50.67	23.29	**72.80**	37.01	13.67	98.43	88.83	**66.14**

4.1, AZB detection is applied to luma (8×8 to 32×32) and chroma (4×4 to 16×6) blocks. It is indicated that our proposal can detect equal or more (in most cases) AZB than [49] does, at all QP values and for all test sequences. The *DA* differences larger than 40% are marked in bold, some are even higher than 70% (for Kimono and Cactus at QP=37). To sum up, proposed algorithm outperforms Sousa's method: detection accuracy for luma blocks is increased by 48.51%, 28.26%, and 8.43% for 8×8, 16×16, and 32×32 AZBs, respectively; meanwhile, for chroma blocks, the increases of detection accuracy for 4×4, 8×8, and 16×16 blocks are 13.43%, 38.14%, and 42.84%, respectively.

The energy conservation based AZB detection method is also evaluated in means of *DA*. The same sequences as shown in Table 4.1 are used for testing and other testing conditions are all the same with Table 4.1. The evaluation results are listed in Table 4.2. It is clear that compared with [49], our proposal can detect more AZBs than Sousa's method can. Lots of the DA differences are larger than 40% and a few are even higher than 70%, for either simple sequences like Vidyo4 or sequences with fast motions like BasketballDrive. As a summary, proposed detection method is superior to Sousa's method: detection accuracy is increased by 58.25%, 39.32% and 24.2% for luma 8x8, 16x16 and 32x32 AZBs, respectively. Meanwhile, DA improvements for chroma 4x4, 8x8, 16x16 blocks are 14.2%, 42.91% and 48.08%, respectively. Compared with M1, DA improvements are 9.77%, 11.06% and 15.76 for luma 8x8, 16x16 and 32x32 AZBs, individually; at the same time, DA improvements are 0.76%, 4.75% and 5.23% for chroma 4x4, 8x8 and 16x16 AZBs. On the other hand, false detection rate should also be evaluated, it is defined as

Table 4.2 *DA* performance comparison of M2 with Sousa's method [49].

Sequences	QP	Sousa's method [49]						Proposed AZB detection method					
		Luma Y			Chroma U/V average			Luma Y			Chroma U/V average		
		8×8	16×16	32×32	4×4	8×8	16×16	8×8	16×16	32×32	4×4	8×8	16×16
flowervase (416×240)	22	59.59	34.87	34.07	95.84	87.83	80.18	90.82	75.82	53.08	99.80	96.98	91.92
	27	64.66	38.81	27.93	97.04	89.65	80.29	93.26	79.68	66.47	99.67	98.80	92.34
	32	62.04	33.53	23.33	97.48	89.04	80.89	96.31	**83.47**	74.21	100.0	99.72	96.38
	37	74.03	49.36	23.11	99.46	95.63	88.61	98.59	87.19	**78.81**	100.0	100.0	99.13
Basketballdrill (832×480)	22	0.24	0.00	0.00	68.92	8.01	0.00	35.83	0.47	0.00	97.94	**83.07**	15.10
	27	2.36	0.00	0.00	79.32	33.39	0.95	**78.02**	3.92	0.00	99.36	**92.27**	**58.49**
	32	2.08	0.00	0.00	84.00	44.41	0.82	**92.44**	25.35	1.32	99.57	**97.74**	**81.39**
	37	17.59	0.35	0.00	96.44	74.93	31.05	**96.76**	**72.68**	20.29	99.38	98.2	**92.70**
Vidyo4 (720p)	22	16.01	5.35	6.19	91.85	47.00	1.99	**81.90**	33.32	9.37	99.96	**99.12**	**74.23**
	27	34.90	6.57	3.78	95.84	70.36	23.02	**92.86**	**58.79**	38.29	99.97	99.75	**95.88**
	32	39.76	6.76	3.27	98.02	77.96	27.41	**98.35**	**76.37**	51.97	99.99	99.83	**98.30**
	37	61.56	26.91	3.90	99.93	96.30	68.05	99.66	**92.64**	72.86	100.0	99.96	99.12
Kimono (1080p)	22	0.68	0.23	0.00	63.92	7.76	0.79	44.49	1.78	1.13	99.00	**85.36**	21.52
	27	6.11	0.27	0.04	80.13	29.13	2.61	**92.70**	20.62	3.89	99.49	**96.99**	61.72
	32	6.67	0.18	0.02	90.71	36.88	2.56	**95.36**	**79.81**	37.32	99.50	**97.72**	**92.82**
	37	42.00	3.48	0.11	99.03	85.66	23.97	**97.99**	**91.44**	77.70	99.71	98.64	**95.08**
Cactus(1080p)	22	0.00	0.00	0.00	45.86	1.45	0.00	1.37	0.00	0.00	**92.61**	60.26	4.38
	27	0.03	0.00	0.00	67.49	18.25	0.05	62.90	0.00	0.00	99.10	**86.01**	38.55
	32	0.04	0.00	0.00	79.48	25.62	0.11	**88.68**	41.93	1.67	99.60	**98.12**	**74.61**
	37	29.23	0.00	0.00	97.80	66.72	17.23	**97.07**	**68.19**	46.06	99.72	98.73	**96.85**
Traffic (2560×1600)	22	1.06	0.07	0.00	57.10	8.07	0.00	**66.86**	1.90	0.00	95.74	**70.07**	13.52
	27	13.32	0.09	0.00	73.92	17.87	2.48	**85.47**	28.88	2.80	98.97	**90.61**	45.20
	32	12.50	0.02	0.00	82.55	33.38	3.07	**95.13**	53.81	23.46	99.68	**98.64**	**78.28**
	37	36.65	3.34	0.00	97.84	71.00	22.94	**98.87**	**75.64**	45.53	99.86	99.35	**95.49**
Average	-	24.29	8.75	5.23	84.99	50.67	23.29	**82.57**	48.07	29.43	99.19	93.58	**71.37**

$$FDR_{N\times N} = \frac{Falsely\ detected\ No.\ of\ N\times N\ AZBs}{Total\ No.\ of\ N\times N\ \ AZBs}\times 100\% \tag{4.28}$$

Table 4.3 shows the values of *FDR* caused by our proposal: the average false detection rate is from 0.06% to 0.25% for all tested sequences which contains different features in texture and motion. In [49], SAD is compared to the theoretical upper limit of N×N transformed coefficient, no false detection will happen and *FDR* in [49] is always 0.

Table 4.3 *FDR* of our proposed AZB detection method (M1).

Sequences	QP	Luma Y			Chroma U/V average		
		8×8	16×16	32×32	4×4	8×8	16×16
flowervase (416×240)	22	0.11	0.00	0.00	0.00	0.00	0.00
	27	0.05	0.03	0.00	0.00	0.00	0.00
	32	0.01	0.03	1.05	0.00	0.00	0.00
	37	0.00	0.00	0.00	0.00	0.00	0.00
Basketballdrill (832×480)	22	0.01	0.00	0.00	0.22	0.00	0.00
	27	0.10	0.00	0.00	0.31	0.00	0.00
	32	0.32	0.00	0.00	0.41	0.00	0.00
	37	0.46	0.12	0.00	0.65	1.44	0.00
Vidyo4 (720p)	22	0.13	0.09	0.25	0.06	0.00	0.00
	27	0.05	0.03	0.14	0.02	0.07	0.00
	32	0.08	0.01	0.00	0.00	0.15	0.70
	37	0.05	0.02	0.00	0.00	0.02	0.86
Kimono (1080p)	22	0.02	0.03	0.00	0.71	0.00	0.00
	27	0.36	0.00	0.04	0.62	0.00	0.00
	32	1.54	0.04	0.00	0.55	1.39	0.00
	37	1.47	0.32	0.01	0.29	1.37	2.67
Cactus (1080p)	22	0.00	0.00	0.00	0.21	0.00	0.00
	27	0.00	0.00	0.00	0.31	0.00	0.00
	32	0.12	0.00	0.00	0.36	0.00	0.00
	37	0.26	0.04	0.00	0.29	0.00	0.00
Traffic (2560×1600)	22	0.05	0.00	0.00	0.32	0.00	0.00
	27	0.12	0.00	0.00	0.27	0.00	0.00
	32	0.14	0.02	0.00	0.25	0.00	0.00
	37	0.13	0.05	0.00	0.13	0.00	0.00
Average	**-**	**0.23**	**0.03**	**0.06**	**0.25**	**0.18**	**0.17**

4.3 Proposed All-Zero Block-based Quadtree Optimization Scheme

This section describes our proposal to fully utilize AZBs of variable sizes to prune PUs that have little contribution. Figure 4.2 presents a schematic diagram of the proposed two-dimensional pruning technique. Vertical pruning examines ME results for 64×64 and 32×32 PUs to constrain PUs in depths 2 and 3. Horizontal pruning skips asymmetric PUs of N×2N and 2N×N when the 2N×2N PU at the same depth can provide good rate distortion (RD) performance. Early termination is introduced in the last subsection.

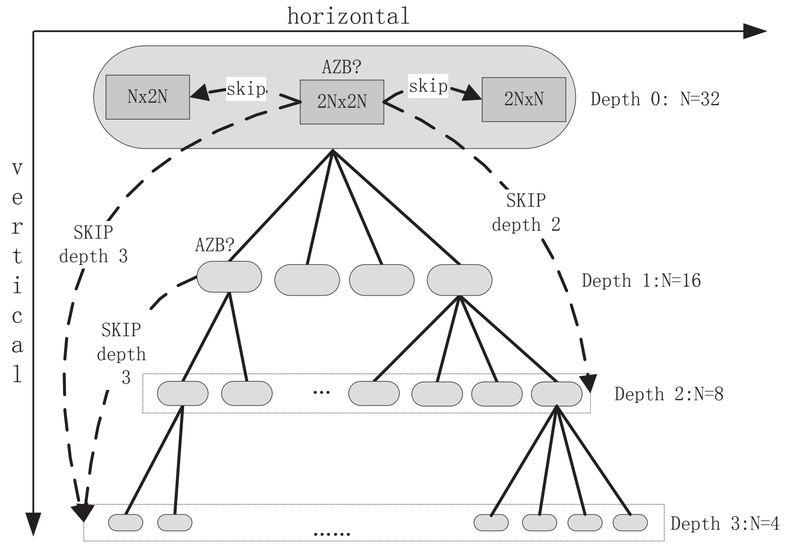


Figure 4.2 Two-dimensional pruning of unnecessary PUs.

4.3.1 All-Zero Block-Based Two Dimensional Quadtree Pruning

Quadtree-based prediction makes motion representation more accurate and adaptive to video signals. It is common for a residual block to be quantized to all-zero. To check whether the residual block of an LCU is an AZB, the sum of absolute quantized transform coefficient (SAQC) is used as a judging criterion. It is defined by

$$SAQC_{64\times 64}=\sum_{k=0}^{k=3}\left\{\sum_{i=0}^{i=31}\sum_{j=0}^{j=31}|T_{32\times 32}\times Q(resi\_bk_{i,j})|\right\}_k \tag{4.29}$$

where *resi_bk* is a 32×32 residual block and *T*×*Q* is the 32×32 DCT followed by quantization. SAQC$_{64\times 64}$ is defined as the sum of absolute coefficients of four quantized residual blocks. The reason for using the largest allowable TU for Eq. (4.29) is that a large transform block can facilitate good preservation of details in quantized signals [4]. If SAQC$_{64\times 64}$ is zero, it is certain that no residual signal remains after quantization and the LCU is an AZB.

For an all-zero LCU, even if partial partitions are skipped, coding performance can be maintained. To verify the validity of this suggestion, simulations are performed to investigate the percentage of all-zero LCUs coded by PUs at each depth. Table 4.4 summarizes the results of tests on 1080p and 4k×2k sequences. The last four columns show the percentages of all-zero LCUs that are coded to depths of 0, 1, 2, and 3, respectively. For the Crowdrun sequences with QP set to 22, no all-zero LCU exists. As expected, the majority of AZBs are coded by PUs in depths 0 and 1. Depths 2 and 3 occupy a very limited proportion, 2.39% and 0.57%, respectively. On the basis of these observations, PUs in depths 2 and 3 can be pruned for all-zero LCUs.

Table 4.4 Percentages of all-zero LCUs coded to depth 0, 1, 2, and 3.

Sequences	QP	Depth 0	Depth1	Depth 2	Depth 3
Crowdrun	22	0.00%	0.00%	0.00%	0.00%
	37	80.08%	12.55%	4.76%	3.89%
Into_tree	27	84.64%	12.2%	2.75%	0.00%
	37	97.42%	2.33%	0.24%	0.00%
Sunflower	32	85.01%	10.89%	2.72%	1.36%
	37	86.88%	8.16%	4.95%	3.18%
Rushhour	22	68.0%	32.0%	0.00%	0.00%
	37	94.71%	3.96%	1.32%	0.00%
Average	**-**	**85.25%**	**11.78%**	**2.39%**	**0.57%**

Non-zero LCUs are further partitioned into four 32×32 CUs. After quantization, 32×32 AZBs are usually present. This implies that further pruning might be possible. To judge whether a quantized 32×32 block is an AZB, a similar criterion is defined as

$$SAQC_{32\times32} = \sum_{i=0}^{i=31}\sum_{j=0}^{j=31} | T_{32\times32} \times Q(resi\_bk_{i,j}) | \tag{4.30}$$

Eq. (4.30) only takes use of 32×32 TU for DCT and Q, and the other two TU sizes, namely 8×8 and 16×16, are not applied. Our verification test to investigate the percentage of 32×32 AZBs coded by depth-3 PUs is performed, and the result reveals that depth-3 only takes less than 1%. Therefore, PUs in depth 3 can be skipped for 32×32 AZBs.

For an LCU, vertical pruning shown by top-to-down arrows in Figure 4.2 can be summarized as follows.

1) Perform 64×64 ME, 32×32 DCT, and quantization. If SAQC in Eq. (4.29) is zero, the LCU is all-zero and PUs in depths 2 and 3 of the entire quadtree are pruned. Otherwise, the LCU is split into four 32×32 CUs.

2) Perform 32×32 ME. If the resulting 32×32 residual block is detected as an AZB using the condition in Eq. (4.20), or if SAQC defined in Eq. (4.30) for this block is zero, prune depth-3 PUs under the corresponding depth-1 tree node.

A large proportion of PUs in Table 3.1 is asymmetric in the form of N×2N and 2N×N. However, when the quantized block of a 2N×2N PU is an AZB, asymmetric PUs of the same depth are found to have little contribution. The judgment that a 2N×2N residual block is all-zero is determined in one of the following two ways: 1) The block is detected as an AZB by using the condition in Eq. (4.20); 2) After N/2×N/2 or N×N DCT and Q, $SAQC_{2N\times 2N}$ defined in Eq. (4.31) is zero. If either condition is satisfied, horizontal pruning of 2N×N and N×2N partitions can be performed.

$$\begin{cases} SAQC_{2N\times 2N} = \sum_{k=0}^{k=15}\left\{\sum_{i=0}^{N/2-1}\sum_{j=0}^{N/2-1}\left|T\times Q(resi\_bk_{i,j})\right|\right\}_k, & N=32,16,8 \\ SAQC_{2N\times 2N} = \sum_{k=0}^{k=3}\left\{\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\left|T\times Q(resi\_bk_{i,j})\right|\right\}_k, & N=4 \end{cases} \tag{4.31}$$

For N larger than 4, a 2N×2N block is all-zero only if 16 sub-AZBs are produced; for N equal to 4, an 8×8 block is all-zero only when four sub-AZBs are produced. Unlike the use of the largest TU in Eq. (4.30), the smallest available TU is used in Eq. (4.31) in

Table 4.5 False rate(%) of judging non-zero 2N×2N residual blocks as AZBs using Eq. (4.29).

	Sequence	QP	64×64	32×32	16×16	8×8
Class C (832×480)	BQMall (complex scene, irregular motion)	22	0.0	0.40	0.70	0.65
		27	0.21	0.94	0.61	0.49
		32	1.06	0.98	0.85	0.46
		37	1.43	0.76	0.67	0.36
	BasketballDrill (fast motion)	22	0.0	1.61	1.82	1.01
		27	1.83	1.41	1.21	0.62
		32	0.92	0.66	0.67	0.36
		37	**1.98**	0.06	0.84	0.21
Class E (720p)	vidyo1 (slow motion)	22	0.09	0.76	0.83	0.52
		27	1.01	0.72	0.47	0.20
		32	0.99	0.96	0.37	0.07
		37	1.42	0.65	0.12	0.03
	vidyo3 (slow motion)	22	0.21	0.53	0.79	0.53
		27	1.82	0.88	0.71	0.48
		32	1.50	0.84	0.68	0.31
		37	1.48	0.48	0.36	0.12
Class B (1080p)	BQTerrace (scene change)	22	0.53	1.65	1.63	1.42
		27	1.30	1.43	0.66	0.71
		32	1.47	0.92	0.51	0.49
		37	1.90	0.87	0.52	0.22
	Kimono (slow motion)	22	0.27	0.50	1.02	1.01
		27	0.49	0.45	0.62	0.42
		32	0.55	0.41	0.74	0.28
		37	0.49	0.49	0.62	0.12
Class A (2560×1600)	PeopleOnStreet (complex scene, irregular motion)	22	0.00	0.01	0.54	1.63
		27	0.09	0.40	1.24	1.28
		32	0.88	1.08	1.71	1.08
		37	1.90	1.51	**1.98**	0.89
	Traffic (middle-level motion)	22	0.39	0.88	1.25	1.15
		27	1.08	0.66	0.95	0.51
		32	1.05	0.92	0.72	0.24
		37	1.31	0.90	0.43	0.10
Average	-	-	**0.92**	**0.80**	**0.83**	**0.56**

that a small transform block will not introduce ring artifacts [4] and it causes fewer false judgments in comparison with using large TUs.

To validate the above condition, we conducted simulations to investigate the rate that Eq. (4.31) falsely judges non-zero residual blocks to be all-zero. Table 4.5 lists the results of tests on sequences with various characteristics. The highest false judging rate is 1.98%, marked in bold font. The average false rates are 0.92%, 0.80%, 0.83%, and 0.56% for 64×64, 32×32, 16×16, and 8×8 PUs, respectively. Since horizontal pruning causes negligible coding loss, overall coding loss due to pruning and false judgment is limited. This was proved by experiments described in Section 5.

4.3.2 All-Zero Block-based Early Termination fro Motion Estimation

With the search point approaching the potential optimal position, SAD decreases gradually during the ME process. When SAD is sufficiently small, it is possible for a residual block to become an AZB after quantization. In this sense, SAD can be utilized for early termination (ET) of the ME process. Since HEVC adopts a complicated interpolation process for fractional ME (FME), ET is designed to speed up both integer ME (IME) and FME.

For squared 2N×2N PUs, when SAD satisfies the conditions in Eq. (20), IME and FME can be terminated to save computation or power. For asymmetric PUs, the following extended conditions are used.

$$SAD_{N\times 2N} = SAD_{2N\times N} \leq 2\times SAD_{N\times N},\ N = 4, 8, 16, 32 \tag{4.32}$$

where $\mathrm{SAD}_{\mathrm{N\times N}}$ is defined in Eq. (4.20). As noted previously, ET is applied to a wide range of inter PUs listed in the third column of Table 3.1.

Figure 4.3 shows the ME searching flow using the proposed early termination technique. Specifically, ET is evaluated at six stages, numbered from 1 to 6 in Figure 4.3. These numbers represent the execution order during ME, with stage 1 being the first processing process. After searching the median MV (the stage 1), the current SAD is compared with the condition in Eq. (4.20) or (4.32) to determine whether the remaining ME stages, stage 2 to 6, can be skipped. Similarly, after the diamond search, ET evaluation is carried out to determine whether the current SAD is small enough to skip the 2-point search or raster search and other processes (the stage 4 to 6). When refinement is completed, ET is evaluated to confirm if FME can be omitted. For large

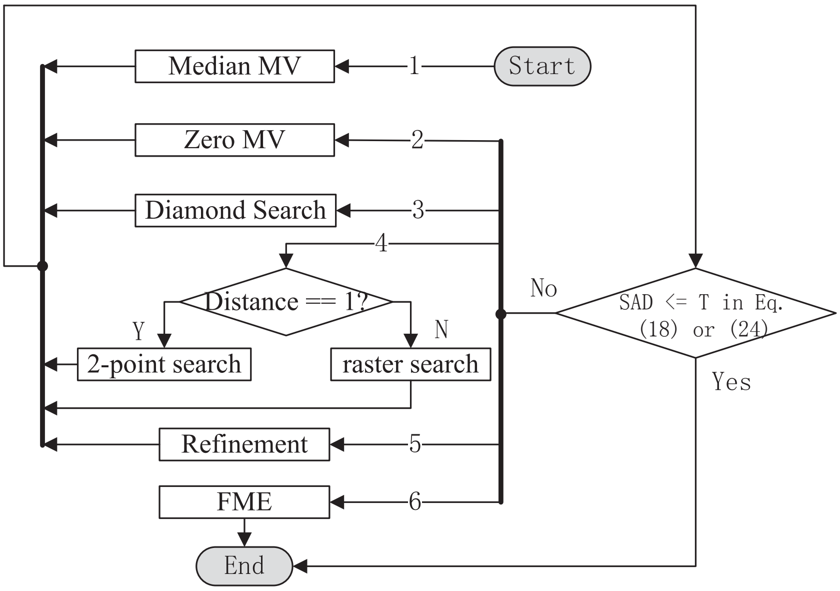


Figure 4.3 ME searching flow with the proposed early termination.

and homogenous regions, ET is usually satisfied when using large partition sizes at an early stage; for complex areas, ET is effective when using small partition sizes. In many cases, IME can find a match that is good enough to skip FME. In these cases, computations and power can be saved.

4.3.3 All-Zero Block-Based Residual Quadtree Pruning

For each quadtree node with size from 8x8 to 32x32, proposed detection methods can be applied after ME, to skip unnecessary DCT and quantization process, as shown in Figure 4.4. Since RQT will split a residual block into sub-blocks and try recursive DCT and quantization for sub-blocks, proposed detection methods skip all the DCT and quantization process if the residual block is judged to be an AZB. Of course, NxN DCT and quantization is skipped only when residual block size is no less than NxN.

The omission of DCT and quantization (Q) steps for the detected AZBs is a straightforward process. For a 2N×2N residual block, supported TUs can be 2N×2N,

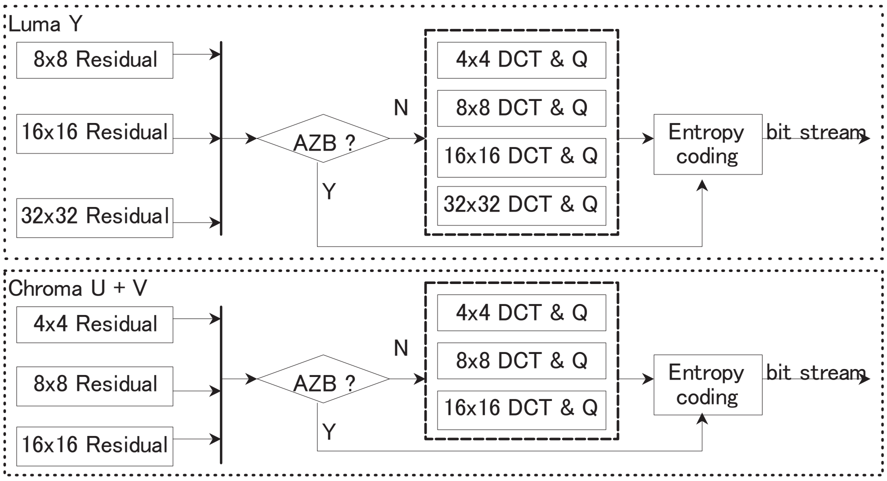


Figure 4.4 Processing flow of Proposed AZB detection methods

N×N and N/2×N/2, as long as the width of the TU ranges from 4 to 32. For the detected AZBs, all allowable TUs are disabled, and DCT and Q can be safely skipped. Another feature of this work is that both luma and chroma residuals are put through to AZB detection, in order to save more computations. As a summary, AZB detection is applied to luma residual block of 8x8 to 32x32 and chroma residual block of 4x4 to 16x16. For both components, the detection condition for a same sized block is identically the same.

Since it is hard to measure computation time which is only spent on DCT and quantization (Q), residual time is used for reduction calculation. Complexity reduction ratio is defined as

$$TR = \frac{\mathrm{Re}\mathit{sidual\ Time}_{\mathrm{Proposed}} - \mathrm{Re}\mathit{sidual\ Time}_{\mathit{Original}}}{\mathrm{Re}\mathit{sidual\ Time}_{\mathit{Original}}} \times 100\% \tag{4.33}$$

Residual time in Eq.(4.33) means the time spent on encoding residuals into bit stream, including DCT, Q, IQ, IDCT and entropy coding. IQ, IDCT and entropy coding are required when residual blocks are not AZBs. It is easy to figure out that *TR* highly depends on the ratio of AZBs, which is further relied on QP and input sequence. Table 4.6 demonstrates coding performance in terms of *BD-PSNR*, *BD-BR* and *TR*. 12

sequences each with 200 frames are tested at QP equal to 22, 27, 32 and 37. The former two evaluators are calculated according to [11].

Table 4.6 Encoding time reduction and coding performance of M1

Sequences		BD-PSRN	BD-BR	TR
416x240	Basketballpass	-0.006	-0.098	-6.36
	BlowingBubbles	-0.002	-0.214	-5.38
	BQSquare	-0.009	-0.130	-6.39
	RaceHorses	-0.001	-0.031	-5.22
	Flowervase	-0.001	-0.218	-14.91
832x480	RaceHorses	0	-0.052	-3.20
	BasketballDrill	-0.005	-0.021	-4.27
	BQMall	-0.001	0.023	-7.43
	PartyScene	0.001	0.010	-4.94
1280x720	Vidyo1	0.005	0.004	-12.30
	Vidyo3	-0.002	0.059	-12.54
	Vidyo4	-0.007	0.027	-11.80
	SlideEditing	0.002	0.289	-12.01
Average	-	-0.002	-0.027	-8.21

4.3.4 Overall Processing Flow

The three proposals are combined to form an optimization scheme for the HEVC encoder. For AZB detection, Eq.(20) detects AZB before DCT so as to skip processing of DCT & Q; Eq.(29)-(31) detects AZB after DCT & Q to ensure every unnecessary partition can be pruned. By the way, if only Eq.(20) is used for AZB detection, reduction in computational complexity will decrease about 20%. The combination of above two AZB detection criteria makes full use of AZB to achieve the best performance in reducing computational complexity. Figure 4.5 shows the processing flow of the proposed scheme.

In depth 0, 64×64 ME is performed to obtain the optimal residual block. DCT and Q in size 32×32 and 16×16 are then performed on the residual block to get transformed blocks. Based on transformed blocks, SAQC defined in Eq.(29) and Eq.(31) is checked to see whether vertical pruning of depth 2 and 3, or horizontal pruning of 64×32 and

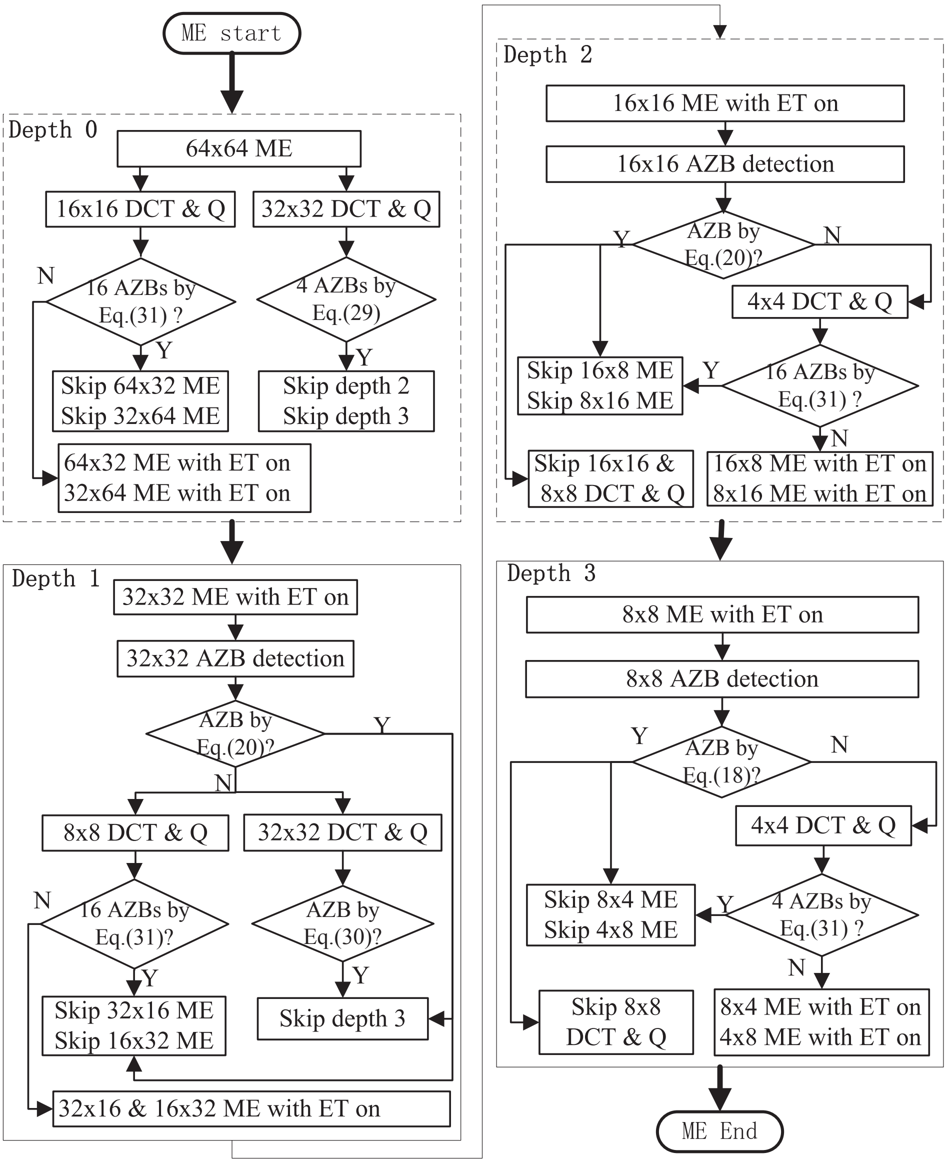


Figure 4.5 Encoding flow of proposed scheme based on AZB utilization.

32×64 can be performed. If horizontal pruning fails, ME with ET enabled is performed for 64×32 and 32×64 PU. Next, the LCU is split into four 32×32 CUs and coding flow goes to depth 1.

In depth 1, 32×32 ME is performed with ET enabled. If generated 32×32 residual block is detected as AZB by Eq. (20), all depth-3 PUs will be pruned and meanwhile, horizontal pruning is enabled. Otherwise, DCT and Q in size 32×32 and 8×8 are executed to check SAQC defined in Eqs.(30) and (31) respectively, to see whether the above pruning can be successful. If horizontal pruning of 32×16 and 16×32 fails, ME with ET enabled is performed for the two partitions. If depth 2 is not pruned in depth 0, each 32×32 CU will be split into four 16×16 CUs and encoding goes to depth 2. Similarly, if depth 3 is not pruned, 16×16 CU will be further split and encoding flow goes to depth 3. In depth 2 and 3, encoding flow shown in the right half of Figure 4.5, is similar to that in depth 1.

For chroma AZB with size from 4×4 to 16×16 in U and V channels, optimization of residual encoding is carried out that is all DCT and Q are disabled shown in Figure 4.4.

4.4 Experiments and Comparison Results

4.4.1 Complexity Analysis of Proposed AZB Detection Algorithm

The computational overhead introduced by the proposed scheme is analyzed in terms of addition, multiplication, shift, and comparison operations. For a luma 2N×2N PU, the calculations required for each proposal are listed in Table 4.7.

As divisions used for AZB detection in Eq. (20) can be pre-calculated for an HEVC encoder, they are not taken into consideration. The remaining computations in Eq. (20) are calculated once per frame and they are also not considered. Table 4.7 shows that AZB detection mainly consumes additions, which are used to calculate SAD. RQT

Table 4.7 Calculations consumed by the proposed AZB detection method M1.

	ADD	MUL	SHIFT	CMP
AZB detection	4×N×N	1	1	1
RQT Pruning	0	0	0	1
Vertical PU Pruning	4 for 64×64; 0 for 32×32	0	0	2
Horizontal PU Pruning	0 for 8×8 12 for others	0	0	4
Early termination	0	1	1	1

pruning shown in Figure 4.4 only needs to compare SAD with the derived threshold in in Eq. (4.20). In vertical pruning, since each quantized result required to calculate SAQC in Eqs. (4.29) and (4.30) can be referred to HM, only four additions are consumed by the 64×64 case, and another two comparisons are needed to check whether the residual block is an AZB. In horizontal pruning, the quantized results required by Eq. (4.31) can also be referred to HM, and at most 12 additions and 4 comparisons are needed. During ME, SAD is calculated by the encoder and ET only needs one multiplication and one shift operation to check whether the residual block will be an AZB.

The proposed scheme demonstrates ultra-low computational complexity and for the worst case when no AZB exists, our experiments which are carried out on representative sequences with various characteristics in resolution and video content at QP equal to 22, 27, 32 and 37 shows that, the percentage of increased computation caused by our AZB detection is always less than 0.65% and averagely it is only 0.388%. Therefore, proposed method is appropriate for implementation in hardware for the sake of saving power. In real cases, the lowest complexity reduction always happens at QP equal to 22 and among all standard test sequences recommended by HEVC [44], the worst reduction using our proposal at QP is 22 is 17.8%.

4.4.2 Optimization Effectiveness

The proposed scheme was implemented into HM 6.0. To evaluate its effectiveness, we carried out experiments using a wide range of video sequences with different characteristics in resolution, temporal motion, and spatial texture [13]. The important parameters in each coding tool and tested number of frames are listed in Table 4.8**.**

Table 4.8 Coding parameters and tested sequences for optimization scheme

Coding parameters	Max CU size=64×64; Max CU depth = 4; IntraPeriod = -1; GOPSize = 4; TUMaxDepthInter = 3; TUMaxDepthIntra = 3; DisableInter4x4 = 1; FastSearch = 1; FEN = 1; RDO = On; InternalBitDepth = 8/10; MRG = 1; AMP = 0; NSQT = 0; SearchRange = 64: Tested frame number : 200 for Class E and below; 100 for Class B/E and 4k×2k

For performance comparison, the evaluation parameters ΔPSNR and ΔBR were used. The overall encoding complexity was simulated by the total encoding time of running a HEVC encoder. The total encoding time of the proposed scheme includes all introduced computational overhead such as AZB detection, and condition checks for pruning PUs.

The coding performance of the proposed scheme versus one of the state-of-the-art methods [11] is shown in Table 4.9, using the low-delay LC configuration [14]. 9 sequences with resolutions ranging from 416×240 to 4k×2k were tested. The contents of the sequences are as follows: Flowervase is captured with a zooming effect; Vidyo1 and Vidyo4 contain scenes of office meetings; SlideEditing contains the motions of modifying slides that are full of characters; Cactus contains irregular motions; Parkscene includes several people riding on bicycles across a green park; Traffic contains dozens of vehicles driving in two directions; *In_to_tree* consists of rotated trees, grass, and a house; and *Old_town_cross* comprises high-angle shots of a town with numerous buildings. These sequences cover a wide range of resolutions, texture, and motions. For all of these sequences, the proposed scheme always achieves higher complexity reductions in the lower bit rate coding scenarios. This is because more AZBs are produced when QP becomes larger, and for these AZBs, asymmetric PUs and PUs in lower depths are successfully pruned. As for bit rate performance, at 3.12%, the rate increment for *SlideEditing* was the worst case observed. For this sequence, the complexity is reduced by 70.46%, which makes the rate increment acceptable. It is also suggested that bitrates have been improved for most sequences when QP is 37. This

Table 4.9 Coding performance comparison with [39] in terms of ΔPSNR, ΔBR, and ΔTime and BD-PSNR and BD-BR.

		Work [39]					Proposed Scheme				
Sequence	QP	ΔPSNR (dB)	ΔBR (%)	ΔTime (%)	BD-PSNR (dB)	BD-BR (%)	ΔPSNR (dB)	ΔBR (%)	ΔTime (%)	BD-PSNR (dB)	BD-BR (%)
Flowervase (zooming) 416×240@30fps	22	-0.07	1.97	-50.28	-0.164	4.44	-0.07	1.75	-41.36	-0.134	3.63
	27	-0.08	1.69	-50.95			-0.10	1.55	-50.00		
	32	-0.11	2.90	-50.98			-0.06	2.38	-57.97		
	37	-0.14	0.75	-50.79			-0.03	-0.72	-65.58		
Average	-	**-0.10**	**1.83**	**-50.75**			**-0.06**	**1.24**	**-53.73**		
Vidyo1 (slow motion) 1280×720 @60 fps	22	-0.07	0.55	-26.60	-0.083	3.30	-0.02	0.10	-41.42	-0.029	1.19
	27	-0.07	1.24	-27.33			-0.01	0.83	-54.79		
	32	-0.06	0.47	-49.32			-0.02	0.08	-67.07		
	37	-0.04	0.23	-50.01			-0.03	-0.63	-73.40		
Average	-	**-0.06**	**0.62**	**-38.31**			**-0.02**	**0.09**	**-59.17**		
Vidyo4 (slow motion) 1280×720 @60 fps	22	-0.03	0.37	-25.93	-0.038	1.78	-0.01	-0.36	-40.65	-0.016	0.82
	27	-0.02	0.50	-26.67			-0.01	-0.33	-50.75		
	32	-0.03	0.71	-50.11			-0.02	0.44	-61.88		
	37	-0.05	0.30	-50.34			-0.04	0.19	-69.59		
Average	**-**	**-0.03**	**0.47**	**-38.26**			**-0.02**	**-0.015**	**-55.72**		
SlideEditing (irregular) 1280×720 @30 fps	22	-0.09	3.35	-44.82	-0.141	5.46	-0.08	3.81	-67.09	-0.085	3.72
	27	-0.14	5.14	-45.62			-0.11	3.86	-70.45		
	32	-0.10	5.02	-45.58			-0.08	1.56	-71.72		
	37	-0.13	4.04	-50.96			-0.08	3.27	-72.60		
Average	-	**-0.12**	**4.38**	**-46.74**			**-0.08**	**3.12**	**-70.46**		
Cactus (middle-level) 1920×1080 @50 fps	22	-0.01	1.02	-26.89	-0.053	2.93	-0.03	0.16	-22.89	-0.040	1.95
	27	-0.04	1.15	-26.41			-0.03	1.01	-37.14		
	32	-0.07	1.05	-46.59			-0.02	0.40	-48.42		
	37	-0.01	0.51	-47.60			-0.03	-0.01	-55.69		
Average	**-**	**-0.03**	**0.93**	**-36.87**			**-0.03**	**0.39**	**-41.04**		
ParkScene (fast motion) 1920×1080 @24 fps	22	-0.02	1.24	-26.20	-0.069	2.61	-0.06	1.34	-26.50	-0.071	2.50
	27	-0.03	1.57	-27.08			-0.04	1.30	-35.25		
	32	-0.06	0.67	-47.05			-0.06	-0.08	-45.83		
	37	-0.04	0.48	-48.32			-0.04	-0.68	-55.66		
Average	-	**-0.03**	**0.99**	**-37.16**			**-0.05**	**0.47**	**-40.81**		
Traffic (fast motion) 2560×1600 @30 fps	22	-0.03	2.05	-25.19	-0.102	3.87	-0.05	1.41	-28.41	-0.070	2.72
	27	-0.09	1.98	-26.25			-0.04	1.30	-41.20		
	32	-0.06	0.54	-26.07			-0.04	0.58	-52.45		
	37	-0.05	0.24	-49.29			-0.04	-0.32	-61.88		
Average	**-**	**-0.05**	**1.20**	**-31.70**			**-0.04**	**0.74**	**-45.98**		
In_to_tree (middle-level) 3840×2160 @30 fps	22	-0.03	-0.08	-25.52	-0.033	2.60	-0.01	-0.18	-18.94	0.001	-1.03
	27	-0.03	1.15	-26.57			-0.01	-1.22	-27.57		
	32	-0.03	0.52	-49.72			0.00	-0.85	-53.69		
	37	-0.02	0.49	-51.44			0.00	-0.75	-70.95		
Average	-	**-0.03**	**0.52**	**-38.31**			**-0.005**	**-0.75**	**-42.78**		
Old_town_cross (slow motion) 3840×2160 @30 fps	22	-0.13	0.07	-21.08	-0.081	0.87	-0.12	0.02	-20.12	-0.005	-1.09
	27	-0.08	-1.12	-25.92			-0.06	-2.48	-28.26		
	32	-0.07	-0.32	-46.95			-0.07	-1.00	-54.80		
	37	-0.10	0.13	-50.54			-0.09	-0.54	-70.46		
Average	**-**	**-0.09**	**-0.31**	**-36.12**			**-0.08**	**-1.00**	**-43.41**		
Total Average	**-**	**-0.06**	**1.18**	**-39.36**	**-0.08**	**3.095**	**-0.044**	**0.394**	**-50.34**	**-0.049**	**1.601**

phenomenon is even more obvious for 4k×2k sequences at any QP. One possible reason is that the pruning of smaller PUs results in a better reconstructed block, which benefits the future encoding. Compared with the original HM 6.0, the proposed scheme reduces complexity for the tested sequences by 40.81% to 70.46%. The average complexity reduction is 50.34%, and the average coding loss in ΔPSNR and ΔBR is less than 0.045 dB and 0.4%, respectively.

The proposed scheme also demonstrates its superiority to [39], which can skip one or two coding depths by comparing the utilization ratios of contiguous depths. For sequences with slower motions such as Vidyo1, Vidyo 4, and SlideEditing, coding time savings by our proposal apparently exceed [39]; 20.86%, 17.46%, and 23.72% gains are further achieved for the above sequences, respectively. Meanwhile, performances of bitrate and PSNR in our proposal are better than that in [39]. For sequences with midlevel or fast motions such as Cactus, Parkscene, and Traffic, rate distortion in the proposed scheme is relatively smaller than that in [39]: the bit rate is reduced by 0.54%, 0.52%, and 0.46%, respectively. Simultaneously, coding complexity savings from our work are also higher. To sum up, our proposal outperforms [39] by an average of 10.98% complexity saving and 0.786% bitrate reduction. Both [39] and proposed method are also evaluated by Bjøntegaard Delta PSNR (BD-PSNR) and Bjøntegaard Delta Bit Rate (BD-BR), calculated according to [18] using four RD points. As disclosed, better BD performance is always achieved for each tested sequence.

A comparison with state-of-the-art [54] is provided in Table 4.10 using four typical sequences with low-delay HE profile [55]. Among the four sequences, BQTerrace and Kimono contain midlevel motions. Scene change also happens for both sequences; the other two sequences contain very fast motions. For evaluation, besides ΔPSNR and ΔBR, BD-PSNR and BD-BR are also used. Table 4.10 indicates that our proposal always achieves greater complexity reductions than [54] when QP is 22. For BQTerrace, [54] is able to reduce the bitrate by 1.19% with 0.02 dB loss in PSNR. However, our proposal can save a further 0.58% in bitrate at the same PSNR. For Kimono, our proposal outperforms [54] by a 1.87% rate saving. For BasketballDrill, [54] only obtains 28.2% complexity saving when QP is 37; on the contrary, our proposal is able to save 53.2%. In summary, our proposal achieves better performance than [54] in terms of BD-BR and BD-PSNR. For average performance of the four sequences, BD-BR loss in our proposal is only 0.165%, which is 1.14% less than [54]. Meanwhile, the average coding complexity reduction of our proposal is approximately 3% higher than [54].

Table 4.10 Coding performance of the proposed scheme in comparison with [54] in terms of BD-PSNR and BD-BR.

		Work [54]					Proposed Scheme				
Sequence	QP	ΔPSNR (dB)	ΔBR (%)	ΔTime (%)	BD-PSNR (dB)	BD-BR (%)	ΔPSNR (dB)	ΔBR (%)	ΔTime (%)	BD-PSNR (dB)	BD-BR (%)
BQTerrace (1920×1080 @60 fps)	22	-0.01	-0.25	-12.6	-0.02	-1.19	-0.03	0.25	-17.8	-0.02	-1.77
	27	-0.02	-0.80	-35.9			-0.05	-0.93	-33.7		
	32	-0.06	-0.53	-54.7			-0.03	-0.94	-49.2		
	37	-0.10	-1.63	-64.9			-0.04	-1.85	-61.2		
Kimono (1920×1080 @24 fps)	22	-0.01	0.23	-29.0	-0.06	1.74	0.0	-0.22	-30.5	-0.004	-0.13
	27	-0.02	0.41	-33.2			0.0	0.04	-37.9		
	32	-0.05	1.02	-54.3			0.003	0.19	-46.6		
	37	-0.06	0.70	-51.7			-0.02	-0.28	-50.9		
BasketballDrill (832×480 @50 fps)	22	-0.03	0.26	-23.7	-0.07	1.84	-0.03	0.35	-29.5	-0.05	1.44
	27	-0.03	0.27	-26.3			-0.04	0.41	-36.7		
	32	-0.08	0.57	-35.3			-0.04	0.52	-44.7		
	37	-0.09	0.88	-28.2			-0.05	-0.13	-53.2		
BasketballDrive (1920×1080 @50 fps)	22	-0.01	0.13	-21.8	-0.06	2.84	-0.01	-0.03	-26.9	-0.02	1.12
	27	-0.02	0.71	-34.8			-0.01	0.54	-36.4		
	32	-0.05	2.27	-48.0			-0.02	0.46	-46.9		
	37	-0.09	2.04	-56.2			-0.03	-0.15	-55.6		
Average	-	-0.045	0.392	-38.16	-0.054	1.31	-0.024	-0.11	-41.10	-0.025	0.165

Figure 4.6 presents comparisons of rate-distortion curves among the original HM, our proposal and referenced works, using QP equal to 22, 27, 32, and 37. Except for SlideEdting, the RD curves of our proposal and HM are closely overlapped. For all sequences, our RD curves always locate above the curves of [39] [54]. The improved RD performances against [39] [54] are partially highlighted by the dashed rectangles in Figure 4.6 (a)-(d). Inside these rectangles, the curves of [39] and [54] are always located below our curves and, it is demonstrated that the proposed scheme achieves better rate-distortion performance than [39] and [54].

4.5 Conclusion

This chapter derives two AZB detection conditions by utilizing sum of absolute difference (M1) and energy conservation before and after DCT (M2). By skipping transform and quantization for AZB, computational burden can be partially eliminated. The detection is performed for luma blocks from 8x8 to 32x32 and chroma blocks from 4x4 to 16x16. Experiments on various video sequences demonstrate that the derived detection method achieves not only high detection accuracy but also accelerates the original residual encoding by up to 14.9%, without performance degradation in terms of picture quality and bit rate.

Moreover, based on AZB detection condition M1, this chapter also proposes an efficient scheme to optimize the quadtree-based prediction and residual encoding. Based on AZB detection condition, the searching process involved in IME and FME can be early terminated once SAD satisfies AZB detection condition. Most significantly, a novel two-dimensional pruning technique based on AZBs, which benefits both homogenous areas and regions with complex motions, is for the first time brought forward to constrain non-contributing PUs. Sufficient statistical data proves that pruning will not introduce obvious coding degradation. Experiments on 416×240 to 4k×2k videos show that our proposal outperforms state-of-the-art methods and reduces complexity for HEVC by up to 73.4% (50.34% on average) while maintaining coding performance. The proposal only requires ultra-low computational complexity and is appropriate to be implemented in hardware to save power.

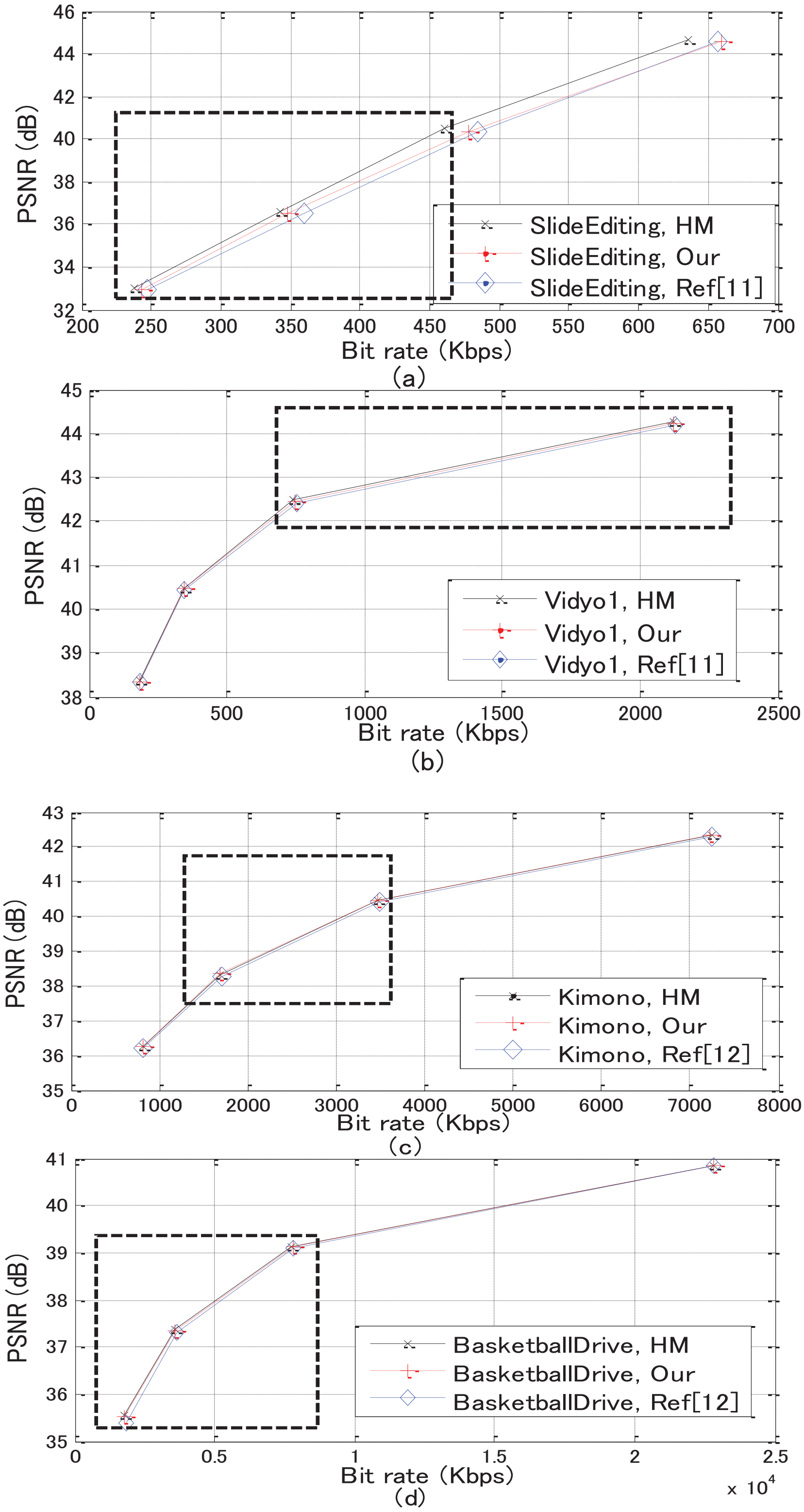


Figure 4.6 Comparisons of rate-distortion curves.

5 Conclusion

The dissertation targets to reduce the computational complexity in video encoders by half while keeping the compression performance in terms of both video quality and encoded bits, so that video data can be compressed efficiently within halt encoding time or under lower power consumption.

Firstly, a mode decision scheme has been proposed for H.264/AVC, in which both prediction modes in intra coding and coding block types in hybrid prediction is reduced largely by means of spatial texture analysis. On the one hand, an edge direction method using Prewitt operator is proposed to constrain intra mode candidates at little hardware cost. On the other hand, a novel homogeneity measurement criteria based on entropy theory is for the firstly time proposed to decide suitable block type and this method achieves better accuracy than previous DCT-based approaches. The above texture analysis approaches can achieve consistent encoding gain for all videos with different motion and spatial features. Average encoding complexity reduction is 52.5% with negligible loss in bit rate and PSNR, which outperforms Liu's method [IEEE Trans. of CSVT 2009].

Secondly, a content adaptive hierarchical decision scheme of variable coding blocks has been proposed for compression videos with high resolutions using HEVC encoder. Considered that traditional methods only utilizes feature in single temporal or spatial domain for complexity reduction, and temporal (spatial) feature can hardly accelerate compression process for videos with very fast motions(complex scene), in proposed approach, multiple features in temporal, spatial and transform domains are extracted and efficiently combined for deciding the most suitable coding blocks. As a result, complexity optimization can be achieved for all kinds of videos containing wide range of characteristics in texture and motion. Comparing with an effective proposal [JCTVC-F092 2011] adopted by HEVC, higher complexity reduction is achieved by our proposed approach. Especially, for videos with complicated scene and very fast motions which are generally recognized as difficult videos for compression, our proposal shows its big advantages.

Thirdly, an ultra low complexity optimization scheme based on All-Zero Block (AZB) detection has been brought forward to reduce power consumption for HEVC encoder. In this proposal, we derive a near-sufficient condition using only residual data to detect variable-sized AZB with size from 32x32 to 4x4 in three video channels. The derived condition only requires little computation of addition and shift operations, and it can detect most AZBs with false detection rate no larger than 0.23%. For detected AZB, DCT and quantization can be skipped without affecting coding performance. In addition, based on the condition, a technique to early terminate motion estimation and a pruning technique to constrain non-contributable prediction blocks are also proposed. Experiments show that our scheme reduces the computational complexity of HEVC by up to 70.46% and 50.34% on average. For sequences with either complicated texture or fast motions, acceleration using proposed approach is effective. In the worst case, even if there is no AZB in video sequence, proposed method only introduce 0.338% complexity of a HEVC encoder. Therefore, proposed method is appropriate for implementation in hardware for the sake of saving power. It also outperforms the latest method proposed by Shen [PCS 2012], in terms of complexity reduction and encoded bits.

Reference

[1] T. Sikora, “MPEG digital video-coding standards”, IEEE Signal Proc. Mag., vol. 14, no. 5, pp. 82-100, Sept. 1997.

[2] M. Liou, “Overview of the px64 kbit/s video coding standard”, Commun. ACM, vol. 34, no. 4, pp. 60-63, Apr. 1991.

[3] K. Rijkse, “H.263: Video coding for low-bit-rate communication”, IEEE Commun. Mag., vol. 34, no. 12, Dec. 1996.

[4]“Draft ITU-T recommendation and final draft international standard of joint video specification”, ITU-T rec. H.264/ISO/IEC 14496-10 AVC, Mar. 2003.

[5] Report of The Formal Verification Tests on AVC, ISO/IEC 14 496-10 ITU-T Rec.H.264 MPEG2003/N6231, Dec.2003.

[6] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, and T. Wedi, “Video coding with H.264/AVC: tools, performance, and complexity”, IEEE Circuits Syst. Mag., vol. 4, no. 1, pp. 7-28, First Quarter, 2004.

[7] Iain E G Richardson, “H.264/MPEG-4 Part 10 White Paper: Intra Prediction”, pp. 1-6, April 2003.

[8] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, “Overview of the H.264/AVC video coding standard”, IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 7, pp. 560-576, Jul. 2003.

[9] M. Liu, “Overview of the px64 kbit/s video coding standard”, Commun. ACM, vol. 34, no. 4, pp. 60-63, Apr. 1991.

[10] Deltev Marpe, Thomas Wiegand and Stephen Gordon, “H.264/MPEG4-AVC Fidelity Range Extensions: Tools, Profiles, Performance, and Application Areas”, IEEE 2005 International Conference on Image Processing (ICIP), Volume 1, pp.593-396, Sep.2005.

[11] K.McCann, B. Bross, S.Sekiguchi and W.J.Han “High Efficiency Video Coding (HEVC) Test Model 2 (HM2) Encoder Description”, document JCTVC-D0502, Oct. 2010.

[12] D. Marpe, H. Schwarz, S. Bosse, B. Bross and P. Helle, “Video Compression Using Nested Quadtree Structures, Leaf Merging, and Improved Techniques for Motion Representation and Entropy Coding”, IEEE Trans. Circuits Syst. Video Technol., vol. 20, no. 12, pp.1676-1685, Dec. 2010.

[13] Mathias Wien, “Variable Block-Size Transforms for H.264/AVC”, IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 7, pp. 604- 613, July. 2010.

[14] K. McCann, B. Bross, I. K. Kim, K. Sugimoto and W. J. Han, “High Efficiency Video Coding (HEVC) Test Model 6 Encoder Description”, document JCTVC-H1002, Feb. 2012.

[15] B. Li, G. J. Sullivan and J. Z. Xu, “Comparison of Compression Performance of HEVC Working Draft 5 with AVC High Profile”, document JCTVC-H0360, Feb. 2012.

[16] Barni, Mauro, “Fractal Image Compression”, Document and image compression (CRC Press) 968: 168–169, May 2006.

[17] Xiangjun, L., Jianfei, C. Robust, “transmission of JPEG2000 encoded images over packet loss channels”, ICME, pp. 947-950, 2007.

[18] G. Bjontegarrd, “Calculation of Average PSNR Differences between RD-curve”, the 13th VCEG-M33 Meeting, Austin, TX, Apr. 2001.

[19] Zhang Kun, Yuan Chun, et al., “A Fast Block Type Decision Method for H.264/AVC Intra Prediction”, The 9th International Conference on Advanced Communication Technology, vol. 1, IEEE Press, Los Alamitos, pp. 673-676, 2007.

[20]Yu-Kun Lin, Tian-Sheuan Chang, “Fast Block Type Decision Algorithm for Intra Prediction in H.264/FRext”, ICIP 2005, vol. 1, IEEE Press, Los Alamitos, pp. 585-588, 2005.

[21] D.Wu, F.Pan, K.P.Lim, S.Wu, Z.G.Li, X.Lin, S.Rahardja, and C.C.ko, “Fast Intermode Decision in H.264/AVC Video Coding”, IEEE Trans. Circuits Syst. Video Technol., vol. 15, no.6, pp.953-958, Jul. 2005.

[22] Zhi Liu, Liquan Shen and Zhaoyang Zhang, “An Efficient Intermode Decision Algorithm Based on Motion Homogeneity for H.264/AVC”, IEEE transactions on Circuits and Systems for Video Technology, VOL.19, NO.1, pp. 128-132, Jan. 2009.

[23] K. Choi, S. H. Park and E. S. Jang, “Coding Tree Pruning Based CU Early Termination”, document JCTVC-092, July 2011.

[24] L. A. Sousa, “General Method for Eliminating Redundant Computations in Video Coding”, Electron Letter, vol.36, no. 4, pp.306-307, Feb. 2000.

[25] X. L. Shen, L. Yu and J. Chen, “Fast Coding Unit Size Selection for HEVC based on Bayesian Decision Rule”, The Picture Coding Symposium (PCS) 2012, pp.453-456, May, 2012.

[26]Andy C. Yu, “Efficient Block-Size Selection Algorithm for Inter-Frame Coding in H.264/MPEG-4 AVC”, ICASSP, pp.169-172, 2004.

[27]Andy Chia, et.al, “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, FEBRUARY. 2008.

[28] Hanli Wang, et al., “An efficient Mode decision algorithm for H.264/AVC Encoding Optimization”, IEEE Transactions on Multimedia, VOL. 9, No.4, JUNE. 2007.

[29] Feng Pan, Xiao Lin, “Fast Mode Decision Algorithm for Intra Prediction in H.264/AVC Video Coding”, IEEE Transactions on Circuits and System for Video Technology, VOL.15. pp. 813-822, Jul. 2005.

[30] Z. Wei, H. Li, K. Ngi Ngan, “An efficient Intra Mode Selection Algorithm for H.264 Based on Fast Edge Classification”, IEEE International Symposium on Circuits and Systems, (ISCAS), pp. 3630-3633, 2007.

[31] Chao-Hsuing Tseng, Hung-Ming Wang, Jar-Ferr Yang, “Enhanced Intra 4x4 Mode Decision for H.264/AVC coders”, IEEE Transactions on Circuits and System for Video Technology, VOL.16. pp. 1027-1032, Aug. 2006.

[32] J. Wang, J. Yang, J. Chen, “A Fast Mode Decision Algorithm and Its VLSI Design for H.264”, IEEE Transactions on Circuits and System for Video Technology, VOL.17, pp. 1414-1422, Oct. 2007.

[33] Hongliang Li, King N.Ngan, “Fast and efficient method for block edge classification”, proceeding of ACM IWCMC 2006 Multimedia over Wireless, pp. 67-72, July.2006.

[34] Heng-Yao Lin, Kuan-Hsien Wu, Bin-Da Liu, and Jar-Ferr Yang “Transform-Based Mode Decision Algorithm for H.264/AVC Intra-Prediction”, IEEE Asia Pacific Conference on Circuits and Systems (APCCAS2008), Macao, China, pp.1344-1347, Dec.2008.

[35] Byung-GYU, et al., “Novel Inter-Mode decision algorithm based on marcoblock (MB) tracking for the P-slice in H.264/AVC video coding”, IEEE Transactions on circuits and systems for video technology, VOL.18, No.2, FEBRUARY. 2008.

[36] Inchoon Choi, Jeyun Lee, “Fast coding mode selection with Rate distortion optimization for MPEG-4 part-10 AVC/H.264”, IEEE Transactions on circuits and systems for video technology, VOL.16, No.12, DECEMBER. 2006.

[37] Frank Bossen, “Common test conditions and software reference configurations”, document JCTVC-D600, Daegu, KR, Jan. 2011.

[38] L. Zhao, L. Zhang, S.Ma, and D.Zhao, “Fast Mode Decision Algorithm for Intra prediction in HEVC”, Visual Communications and Image Processing, pp.1-4, Nov.2011.

[39] J.Leng, L.Sun, T.Ikenaga, and S.Sakaida, “Content Based Hierarchical Fast Coding Unit Decision Algorithm for HEVC”, International Conference on Multimedia and Signal Processing, pp.56-59, 2011.

[40] G.F.Tian, T.R.Zhang, and S.Goto, “A macroblock homogeneity detection method and its application for block size detection in H.264/AVC”, The Journal of the Institute of Image Electronics Engineers of Japan, Vol. 39, No. 5, pp.672-691, Sep. 2010.

[41] M.H.Zhou and V.Sze, “Evaluation of Transform Unit (TU) Size”, document JCTVC-C056, Oct.2010.

[42]Guifen Tian, Tianruo Zhang, Takeshi Ikenaga, Satoshi Goto, “A Block Type Decision Algorithm for H.264/AVC Intra Prediction Based on Entropy Feature”, IEEE Asia Pacific Conference on Circuits and Systems, Macao, China, pp.1348-1351, Nov.2008.

[43] https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-4.0.

[44]JCT-VC, “Common test conditions and software reference configurations”, JCTVC-C500, JCT-VC Meeting, Guangzhou, Oct.2010.

[45] G.Bjontegaard, “Calculations of Average PSNR Differences between RD-Curves”, Doc. VCEG-M33, Apr. 2011.

[46] JCT-VC, “WD1: Working Draft 1 of High-Efficiency Video Coding”, JCTVC-C403, JCT-VC Meeting, Guangzhou, Oct.2010.

[47] F.Bossen, “Common test conditions and software reference configurations”, document JCTVC-F900, Jul.2011.

[48] K. Choi, S. H. Park and E. S. Jang, “Coding tree pruning based CU early termination”, document JCTVC-F092, July 2011.

[49]L. A. Sousa, “General Method for Eliminating Redundant Computations in Video Coding”, Electron. Letter, vol. 36, no. 4, pp. 306-307, Feb. 2000.

[50] Y. H. Moon, G. Y. Kim and J. K. Kim, “An Improved Early Detection Algorithm for All-Zero Blocks in H.264 Encoding”, IEEE Trans. Circuits Syst. Video Technol., vol. 15, no. 8, pp. 1053-1057, Aug. 2005.

[51] Y. M. Lee and Y. Y. Lin, “Zero-Block Mode Decision Algorithm for H.264/AVC”, IEEE Trans. Image Processing, vol. 18, no. 3, pp. 524-533, Mar. 2009.

[52]Q. Liu, Y. Q. Huang, S. Goto, T. Ikenaga, “Hardware-Oriented Early Detection Algorithms for 4 × 4 and 8 × 8 All-Zero Blocks in H.264”, IEICE Trans. Fundamentals, vol. E92-A, no. 4, pp. 1063-1071, Apr. 2009.

[53]A. Fuldseth, G. Bjøntegaard, M. Budagavi and V. Sze, “Core Transform design for HEVC”, document JCTVC-G495, Nov.2011.

[54]X. L. Shen, L. Yu and J. Chen, “Fast Coding Unit Size Selection for HEVC based on Bayesian Decision Rule”, The Picture Coding Symposium (PCS) 2012, pp. 453-456, May. 2012.

[55] https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-6.0.

[56] I.E.G. Richardson, “H.264 and MPEG-4 video compression”, John Wiley & Sons Ltd., 2003.

[57] T. Koga, K. Linuma, A. Hirano, Y. Lijima, and T. Ishiguro, “Motion-Compensated inter frame coding for video conferencing”, in Proc. Nat. Telecommunications Conf., pp. G5.5.1- pp.G.5.3.5, Nov./Dec, 1981.

[58] M.Yang, W.Wang, “Fast Macroblock mode selection based on motion content classification in H.264/AVC”, in Proc. IEEE Conf. ICIP, vol.2, pp. 741-744, Oct. 2004.

[59] G. J. Sullivan and J. R. Ohm, “Meeting Report of the first meeting of the Joint Collaborative Team on Video Coding (JCT-VC)”, document JCTVC-A200, Dresden, DE, 15-23, April, 2010.

[60] Tu, Y. K., Yang, J. F., Sun, M. T., 2007, “Rate-Distortion Modeling for Efficient H.264/AVC Encoding”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 17, pp. 530-543.

[61] “Test Model under Consideration”, document JCTVC-A205.doc, Dresden, April, 2010.

[62] D. Marpe, H. Schwarz, and T. Wiegand, “Novel Entropy Coding Concept”, Joint Collaborative Team on Video Coding, document JCTVC-A032.doc, Dresden, Germany, April, 2011.

[63] Nguyen, H. Schwarz, H. Kirchhoffer, D. Marpe, and T. Wiegand, “Improved context modeling for coding quantized transform coefficients in video compression”, in Proc. Picture Coding Symposium (PCS), Krakow, Poland, May.2012.

Publication

Journal

[1] **Guifen Tian,** Xin Jin, and Satoshi Goto, "All-Zero Block-Based Optimization for Quadtree Structured Prediction and Residual Encoding in High Efficiency Video Coding", IEICE Transaction on Fundamentals of Electronics, Communications and Computer Sciences, vol.E96-A, No.4, pp.769-779, Apr. 2013.

[2] **Guifen Tian,** Xin Jin, and Satoshi Goto, "Content Adaptive Hierarchical Decision of Variable Coding Block Sizes in High Efficiency Video Coding for High Resolution Videos", IEICE Transaction on Fundamentals of Electronics, Communications and Computer Sciences,Vol.E96-A,No.4, pp.780-789, Apr. 2013.

[3] **Guifen Tian,** Tianruo Zhang, and Satoshi Goto, "A macroblock homogeneity detection method and its application for block size decision in H.264/AVC", The Journal of the Institute of Image Electronics Engineers of Japan, vol. 39, no. 5, pp. 672-681, Sep. 2010.

[4] Tianruo Zhang, **Guifen Tian**, Takeshi Ikenaga, and Satoshi Goto, "High throughput VLSI architecture of a fast mode decision algorithm for H.264/AVC intra encoding", IEICE Transaction on Fundamentals of Electronics, Communications and Computer Sciences, vol.E91-A, no.12 pp.3630-3637, Dec. 2008.

International Conference

[1] **Guifen Tian** and Satoshi Goto, "An Optimization Scheme for Quadtree-Structured Prediction and Residual Encoding in HEVC", IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), pp.547-550, Kaohsiung, Taiwan, Dec.3-5, 2012.(best paper award)

[2] **Guifen Tian** and Satoshi Goto, "All Zero Block Detection Algorithm for Residual Quadtree Encoding in HEVC", International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC), Sapporo, Japan, July 15-18, 2012.

[3]**Guifen Tian** and Satoshi Goto, "Content adaptive prediction unit size decision algorithm for HEVC intra coding", Picture Coding Symposium (PCS), pp. 405-408, Krakow, Poland, May, 2012.

[4] **Guifen Tian**, Tianruo Zhang, Takeshi Ikenaga, and Satoshi Goto, "A fast hybrid

decision algorithm for H.264/AVC intra prediction based on entropy theory", International Multimedia Modeling Conference (MMM), pp. 85-95, Sophia-Antipolis, France, LNCS 5371, Jan. 2009.

[5] **Guifen Tian**, Tianruo Zhang, Xianghui Wei, and Satoshi Goto, "A block type decision algorithm for H.264/AVC intra prediction based on entropy feature", IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), pp. 1348-1351, Macao, China, Nov. 2008.

[6] **Guifen Tian**, Tianruo Zhang, Xianghui Wei, Wenqi You, Takeshi Ikenaga, and Satoshi Goto, "An efficient fast mode decision algorithm for H.264/AVC intra prediction", International Congress on Image and Signal Processing (CISP), vol. 1, pp.411-415, Sanya, China, May 2008.

[7] Tianruo Zhang, **Guifen Tian**, and Satoshi Goto, "A frequency-based fast block type decision algorithm for intra prediction in H.264/AVC high profile", IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), pp. 1292-1295, Macao, China, Nov. 2008.

[8] Tianruo Zhang, Shen Li, **Guifen Tian**, Takeshi Ikenaga, and Satoshi Goto, "High throughput VLSI architecture of a fast mode decision algorithm for H.264/AVC intra prediction", International Conference on Communications, Circuits and Systems (ICCCAS), pp. 1245-1249, Xiamen, China, May 2008.

[9] Tianruo Zhang, **Guifen Tian**, Takeshi Ikenaga, Satoshi Goto, "A Novel Fast Block Type Decision Algorithm for Intra Prediction in H.264/AVC High Profile", The 23rd International Technical Conference on Circuits/Systems, Computers and Communications, Shimonoseki, Japan, pp.1-4, Jul.2008.

[10] Xianghui Wei, Wenming Tang, **Guifen Tian**, Takeshi Ikenaga, Satoshi Goto, "A Hardware-Oriented High Precision Motion Vector Prediction Scheme for MPEG-2 to H.264 Transcoding", CISP, Hainan, China, Vol.1, pp.278-282, May. 2008.

[11] Xianghui Wei, Wenming Tang, **Guifen Tian**, Yan Zhuang, Takeshi Ikenaga, Satoshi Goto, "Bandwidth Reduction Schemes for MPEG-2 to H.264 Transcoding Design", The 16th European Signal Processing Conference (EUSIPCO 2008), Lausanne, Switzerland, Aug.2008.

[12] Xianghui Wei, Wenming Tang, **Guifen Tian**, Takeshi Ikenaga, Satoshi Goto, "A Low Bandwidth Integer Motion Estimation Module for MPEG-2 to H.264 Transcoding", IEEE Asia Pacific Conference on Circuits and Systems(APCCAS), Macao, China, pp. 1470-1473, Nov.2008.

Domestic Conferences

[1] **Guifen Tian**, Tianruo Zhang, Wenqi You, Takeshi Ikenaga, Satoshi Goto, “A Fast Mode Decision Algorithm for H.264/AVC Intra Prediction”, 電子情報通信学会2008総合大会, Kitakyushu, Japan, A-16-2, pp.282, Mar. 2008.

[2] **Guifen Tian**, Tianruo Zhang, Takeshi Ikenaga, Satoshi Goto, “A Fast Block Type Decision Algorithm for H.264/AVC Intra Prediction”, 2008年電子情報通信学会通信ソサイエティ大会, 通信講演論文集2, BS-12-19, pp.S-153, Sep. 2008.