Study on key technologies for video coding standard beyond H.265/HEVC

A Dissertation Submitted in Candidacy for the Ph. D. Degree
Department of Electrical and Electronic Engineering,
System Innovation Engineering,
Graduate School of Advanced Technology and Science

University of Tokushima, Japan

JIANG XIANTAO

September 2016

Abstract

H.265/HEVC is the new next generation video compression standards, which is developed by the ITU-T video coding experts group (VCEG) together with the ISO/IEC moving picture experts group (MPEG). The final drafting version of H.265/HEVC is released in 2013. Afterwards, the standard is continually extended to support more new advanced video applications which include high resolution, scalable or multiview video applications. As evaluated by many researchers, the H.265/HEVC standard can achieve much more coding efficiency compared with the previous video compression tools. However, the complexities of the corresponding algorithms have increased the difficulty of implementation. To implement the encoder only in the way of software will exhaust the current general purpose processor based device, especially for the high resolution, multiview video applications etc. Therefore, high speed, application specified hardware has been acknowledged as a good way of implementation for the H.265/HEVC encoding. This paper studies on the key technology of the next generation video coding.

In this paper, focusing on the spatial motion vector prediction(SMVP) and temporal motion vector prediction (TMVP), parallel improvement for spatio-temporal prediction algorithms are presented, which can remove the dependency between prediction coding units and neighboring coding units. Using this proposal, it is convenient to process motion estimation in parallel, which is suitable for different parallel platforms such as multi-core platform, compute unified device architecture (CUDA) and so on.

In order to improve the encoding efficiency in HEVC. Therefore, an efficient prediction motion vector (PMV) selection algorithm with adaptive search range (SR) adjustment algorithm is proposed. Firstly, an improved PMV selection algorithm is proposed to further improve the coding efficiency of inter prediction. The MVs surrounding the coding unit are used to get the accurate PMV. After that, the SR is adjusted adaptively according to the depth of coding unit in motion estimation process, which can reduce the computaional complexity in motion estimation. This proposed method can achieve a better tradeoff between the encoding efficiency and

encoding complexity.

In order to balance the trade-off between encoding efficiency and encoding complexity in HEVC inter prediction. A high efficiency CU (coding unit) size decision algorithm is proposed for HEVC inter coding, which contains two methods: CU size termination decision method and CU size skip decision method. The CU splitting is modeled as a binary classification problem based on Naive Bayes model. The difference from previous works is that residual flag in inter-coded CU is used to predict as a feature. The offline learning method is used to obtain the statistical parameters. Further, in order to improve the accuracy of classification, an improved CU size decision algorithm is proposed based Markov Random Field (MRF) model, which uses the encoding information of neighboring CU to the prior probability of CU splitting and no-splitting. This method can reduce the encoding complexity significantly.

The proposed approaches can enhance the performance of video coding beyond HEVC further, Moreover, for multi-view video coding, scalable video coding and screen content coding applications, this methods are also applicable.

Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Background

Recently, high-definition ($4K\times2K$, $8K\times4K$) video applications are widely used. The video compression technology presents the great challenge. Furthermore, many kind of difference video applications continuously appears with the internet and memory technology. Nowadays, digital video broadcasting, wireless mobile video service, remote monitoring and medical imaging have entered people's life. Thus, on April 2010, the Joint Video Team (JVT) that is released by the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG) plan to develop the new next generation video compression standard-H.265/HEVC [1].

On April 2010, the first JCT-VC conference is held in Dresden Germany. The name of the next generation video compression standard High efficiency video coding (HEVC) has been established. The core goal of HEVC is to double compression efficiency based H.264/AVC high profile [2]. HEVC is allowed to increase the coding complexity in encoder side, while the encoding efficiency is improved.

On October 2010, the first HEVC draft specification is published in the third JCT-VC conference, meanwhile, the formal HEVC reference software model (HM) is published. The first branch of HM1.0 is released on January 2011.

On February 2012, the real milestone HEVC specification committee draft is published by JCT-VC, which means this work has made a significant progress, and subsequent HEVC special draft need to be further perfected.

On April 2013, H.265/HEVC has been formally accepted as an international standard by ITU-T, and H.265/HEVC standard is released on the ITU-T website. On November 2013, the H.265/HEVC standard is pushlished by ISO/IEC.

After the standard released, the relevant standard further work is still ongoing. the existing work of JCT-VC mainly focuse on the extension of HEVC: Scalabe HEVC (SHVC)[3], 3D-HEVC [4] and HEVC Screen Content Coding (SCC) [5].

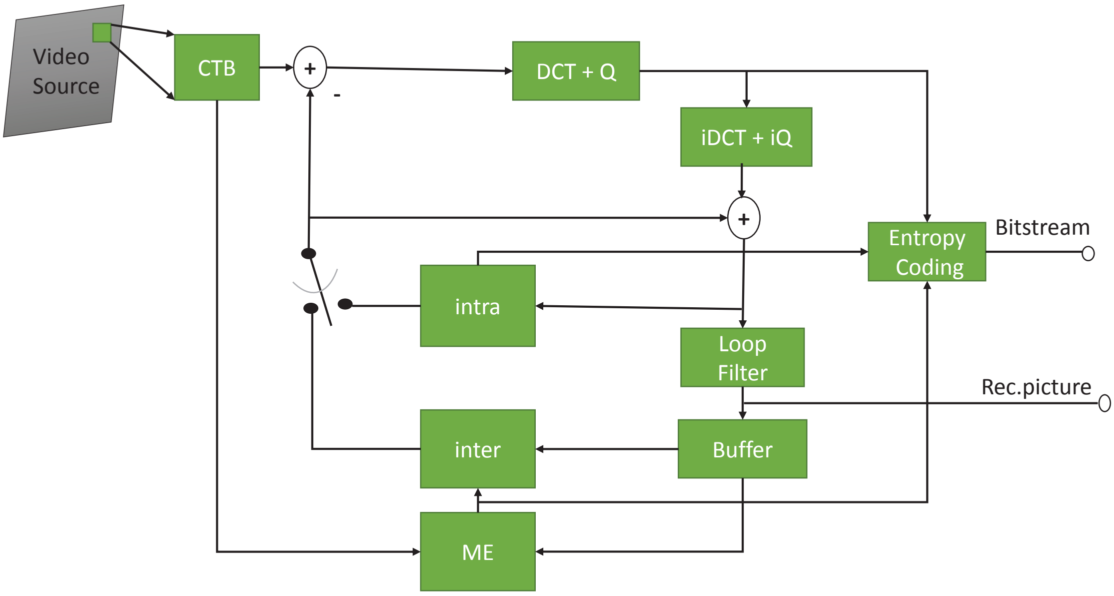

Figure 1.1 HEVC hybrid video coding structure diagram

1.2 Video coding framework

Similar to the previous video compression standards of ITU-T and ISO/IEC MPEG, the hybrid video coding scheme has been used in H.265/HEVC. While the structure has not been changed, the algorithms represented by the building blocks have been refined and the applicable configuration for the algorithms has become more and more flexible over the last 25 years. The basic structure of HEVC scheme is shown as Fig.1.1(CTB: coding tree block; ME: motion estimation; DCT: discrete cosine transform; Q: quantization). The hybrid structures include: intra prediction, inter prediction, transform, quantization, in-loop filtering and entropy coding.

- Intra prediction

This module is mainly used to remove the spatial correlation of the image. The information of the reconstructing encoded unit is used to predict the current coding unit to remove the spatial redundancy and improve the efficiency of image coding. Compared with H.264/AVC, H.265/HEVC supprots more intra prediction modes.

- Inter prediction

This module is mainly used to remove the temporal correlation of the image. The reference frames are used to obtain the motion vector(MV) of the current coding unit, so as to remove temporal redundancy and improve the compression efficiency.

In H.265/HEVC, inter prediction can support uni-directional and bi-directional prediction.

- Transform and quantization

This module is mainly to remove frequency domain correlation using the residual data by transform and quantization, and this technology is lossy compression. Through the transform coding, the image signal is from the time domain to the frequency domain, and the energy is concentrated in low frequency area. Quantization module can reduce the dynamic range of video coding. Moreover, this two processes together can reduce the computational complexity in HEVC.

- In-loop filtering

In-loop filtering technology is used in H.265/HEVC, which includes two modules: deblocking filter (DF) and sample adaptive offset (SAO). DF module is used to reduce blocking effect, and adaptive pixel compensation is used to improve the ringing effect. SAO module is used to reduce the prediction residual of the subsequent coding pixels, and improve the quality of video effectively.

- Entropy coding

In entropy coding module, the code control data, quantization transform coefficient, frame prediction and motion data are coded as a binary stream for storage or transport. The output data of this module is original video compressed code stream. Context-based adaptive binary arithmetic coding (CABAC) technology is used in H.265/HEVC.

1.3 Motivation of thesis

Compared with H.264/AVC, encoding performance of H.265/HEVC has been improved significantly. However, in the future, video data will account for half of the network traffic; high-definition video applications will be applied widely; and mobile terminal equipments need to deal with a lot of video data. Therefore, there are mainly three requirements for video compression: parallel processing, high compression capability, and low computational complexity.

- Parallel processing

It is a great challenge for the computational capabilities of processor and real-time processing in H.265/HEVC [8]. For the high computational complexity, it can not only depend on the ability of the processor to calculate. With the development of multi-core processor architecture, parallel processing in a multi-core environment can multiply decoding speed. However, there are data dependence between coding unit in HEVC. The dependence exists in intra and inter prediction. (1) intra prediction dependence: The current CTB depend on the pixel and mode information of the left, top Left, top, and top right CTB. (2) inter prediction dependence: The MV of the current CTB is predicted by the left, top Left, top, and top right CTB. Thus, how to eliminate dependencies between coding unit is a very important question.

- High compression capability

Compression capability is the basic fundamental driving force behind the adoption of video compression technology [9]. AVC had about twice the compression capability of MPEG-2, and H.265/HEVC had also about twice the compression capability of H.264/AVC. HEVC can compress video about twice as much as AVC without sacrificing quality. In order to meet the needs of people for high resolution video, how to improve coding efficiency further is a very important question.

- Low computational complexity

The computational complexy has been increased with the coding efficiency improved [10]. Compared with H.264/AVC, HEVC computational complexity is mainly embodied in the following respects: (1) there are more modes in intra prediction. (2) the block partitioning methods are more diverse. (3) the transform unit (TU) has been introduced in HEVC. HEVC computational complexity is dozens of times before codec. Therefore, how to reduce coding compelxity is a very important question.

In this work, three approaches are proposed for the key technology for the next generation of efficient video coding. Firstly, a spatio-temporal prediction algorithm is proposed to improve the parallelism of motion estimation in HEVC. Secondly, an efficient motion vector prediction selection algorithm is presented. The MVs surrounding the coding unit are used to get the accurate PMV. After that, the search range is adjusted adaptively in motion estimation process. Thirdly, a fast CU size decision algorithm is presented. The proposed algorithm consists of CU termination and CU skip methods to reduce the redundant computing of inter prediction in HEVC.

1.4 Thesis outline

This paper is organized as follows. The current chapter gives a brief introduction on history of the coding standards and coding framework.

The chapter 2 gives the overview of HEVC in detail, and reference software HM is introduced briefly.

In chapter 3, a spatio-temporal prediction algorithm is proposed to improve the parallelism of motion estimation in HEVC.

In chapter 4, an efficient motion vector prediction selection algorithm is presented.

In chapter 5, a fast CU size decision algorithm is introduced.

In chapter 6, conclusions and the future work.

Chapter 2 Overview of HEVC

2.1 Overview of HEVC

The High Efficiency Video Coding (HEVC) standard is designed along the successful principle of block-based hybrid video coding. This chapter describes the more details about the basic encode modules of HEVC.

2.1.1 Coding structures

- Slice

A slice is data struct that can be decoded independently from slices of the same picture, in terms of entropy coding, signal prediction and residual signal reconstruction [11]. A slice can either be the entire picture or a region of a picture, which is not necessarity rectangular. It consists of a sequence of one or more slice segments starting with an independent slice segment and containing all subsequent dependent slice segments that precede the next independent slice segment within the same access unit.

A slice segment consists of a sequence of coding tree units (CTUs). An independent slice segment is a slice segment for which the value of the syntax elements of the slice segment header are not inferred from the values for a preceding slice segment. A dependent slice segment is a slice segment for which the value of some syntax elements of the slice segment header are inferred from the values for the preceding independent slice segment in decoding order.

Fig.2-1 shows the relation between slice and slice segment(SS). The picture is divided into two slices. The first slice contains one independent SS and two dependent SSs, and the second slice contains one single independent SS.

- Tile

A tile is a rectangular region containing an integer number of coding tree units in coding block raster scan [12]. The tile scan order is a specific sequential ordering of coding tree blocks partitioning a picture in which the coding tree blocks are ordered consecutively in coding tree block raster scan in a tile, whereas tiles in a picture are

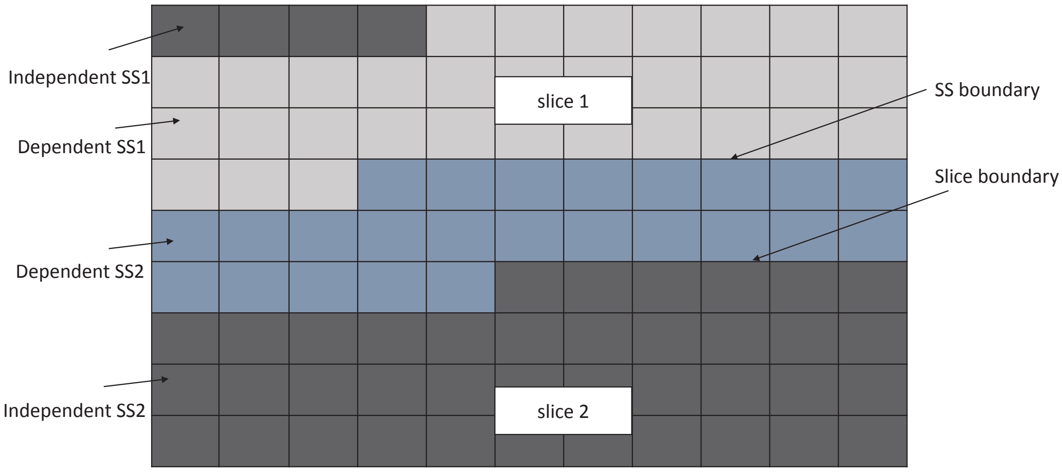

Figure 2.1 Example of slice and slice segments

ordered consecutively in a raster scan of the tiles of the picture.

A tile may consist of coding tree units contained in more than one slice. Similarly, a slice may consist of coding tree units contained in more than one tile.

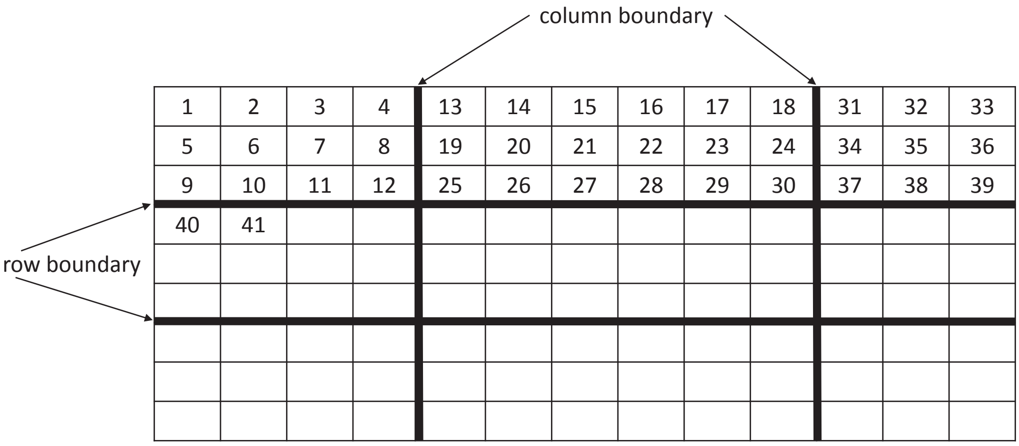

Figure 2.2 The partitioning of tile

In general, all tiles are processed according to the scanning order, and the CTUs within tile are encoded according to the scanning order. Fig.2.2 shows the tile partitioning which is 3×3 partitioning, and the picture is divided into nine tiles.

- CTU pratitioning

Picture are divided into a sequence of coding tree unit (CTUs) [13]. The CTU concept is broadly analogous to that of the mackblock in H.264/AVC. For a picture that has three sample arrays, a CTU consists of an $N \times N$ block of luma coding tree blocks(CTB) together with two corresponding blocks of chroma CTBs. The luma

CTB and the two chroma CTBs together with associated syntax form a CTU. In HEVC, the maximum size of luma CTB in a CTU is 64×64. The CTU is the basic processing unit used in the standard to specify the decoding process. Larger size CTU can achieve better coding efficiency, while it may increase the computational complexity. Thus, it can achieve a better trade-off for the targeted application by choosing the CTU size.

- Coding unit (CU) structures

The coding unit (CU) is a square region, represented as the leaf node of CTU, which shares the same prediction mode: intra, inter or skipped. The quadtree partitioning structure allows recursive splitting into four equally sized nodes. This process gives a content-adaptive coding tree structure comprised of CUs, each of which may be as large as the CTU or as small as 8×8. In a CTU, the split flag (*split_cu_flag*) is used to indicated whether CU is split into for equally-sized CU. If the CTU is split, for each of the resulting CU, another flag is transmitted specifying whether the block represents a CU or further split into four block. When the CU size is 8×8, no splitting flag is transmitted for the encoding block. However, the computation complexity of CU partitioning is high, and a low-complexity encoder is used that does not choose coding block smaller than a larger size.

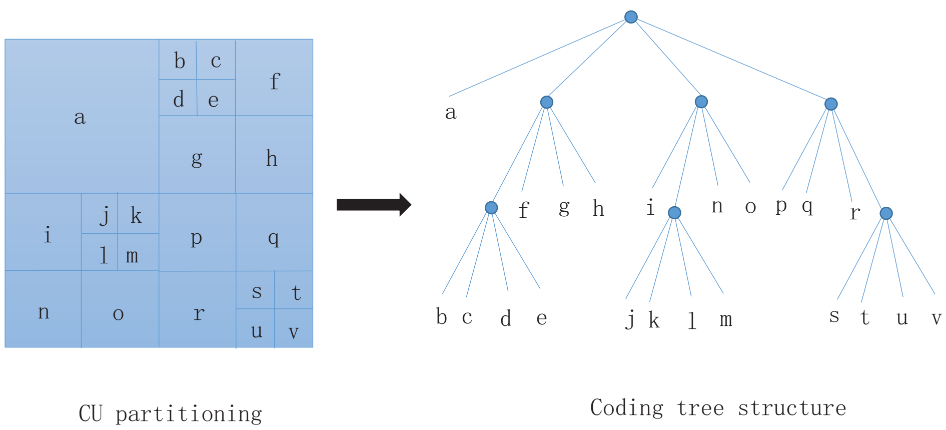

Figure 2.3 Example for the CU partitioning

CU partitioning and the coding order is referred to as z-scan and is illustrated in Fig.2.3. Assume that the size of coding unit CU_d is $2N \times 2N$, and the depth is d. If the value of *split_cu_flag* is 0, CU_d is not split. On the contrary, CU_d is split into four equally-sized CU_{d+1}, and the size of CU_{d+1} is $N \times N$, and the depth is $d+1$. Compared with H.264/AVC, thre are some advantages using the current coding structure as follow:

(1). the size of coding unit in H.265/HEVC is bigger than the macroblock in H.264/AVC. For the smooth region, it can reduce the number of bitrate by using the large coding size, and it can improve the coding efficiency.

(2). by choosing the suitable CTU size and the maximum depth of CU, the coding structure can be optimized significantly.

(3). the coding structure can be expressed easily by the size of CTU, the maximum depth of CU, and the split flag.

- Prediction unit (PU) structures

The prediction unit (PU) is a region, defined by partitioning the CU, on which the same prediction is applied. In general, the PU is not restricted to being square in shape, in order to facilitate partitioning which matches the boundaries of real objects in the picture.

All prediction mode is defined by the PU, and the informations about prediction are included in the PU, such as inter prediction mode, the inter partitioning, the motion vector, the index of reference frame.

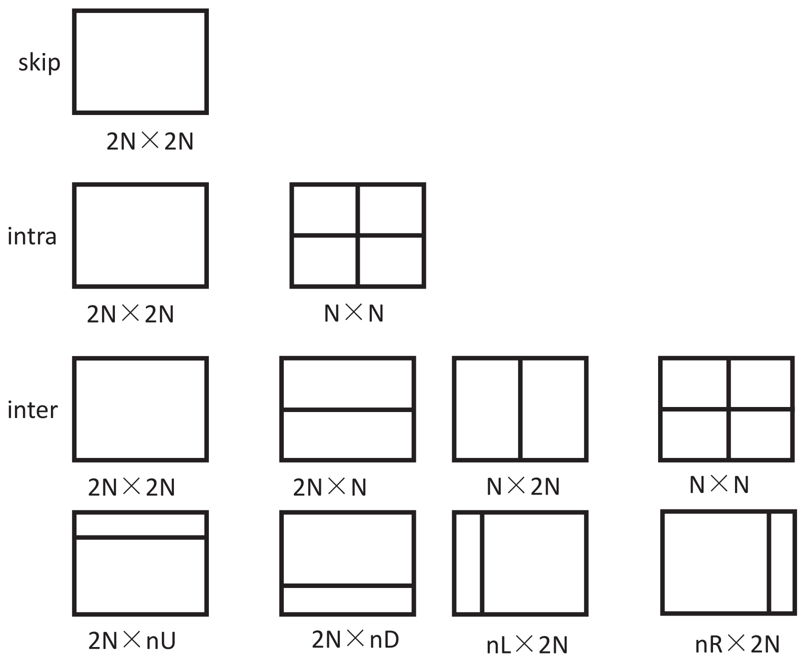

Figure 2.4 Prediction mode for the PU

The prediction mode of the CU with $2N \times 2N$ partitioning is illustrated in

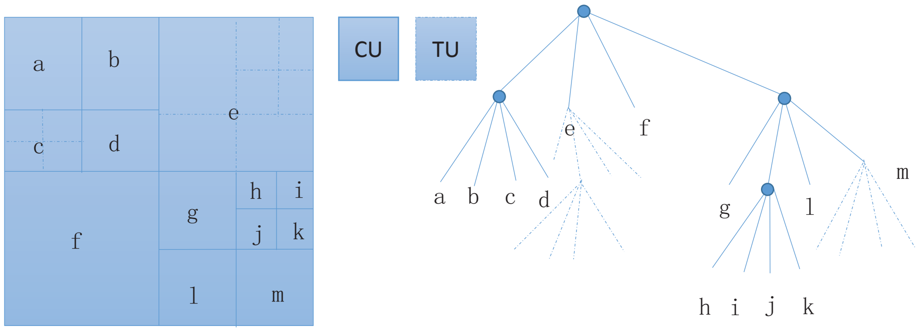

Figure 2.5 Example of transform tree structure within CU

Fig.2.4. For the CU with $2N \times 2N$ partitioning, there are two prediction modes in intra prediction: $2N \times 2N$ and $N \times N$, and eight prediction modes in inter predcition that includes four symmetric modes ($2N \times 2N$,$2N \times N$, $N \times 2N$,$N \times N$) and four asymmetric modes ($2N \times nU$,$2N \times nD$,$nL \times 2N$,$nR \times 2N$). Skip mode is one of mode in inter prediction. When the motion information is only the index of motion parameter set and the coding residual information does not need coding, the prediction mode is skip.

- Transform unit (TU) structures

The transform unit (TU) is a square region, defined by quadtree partitioning of the CU, which shares the same transform and quatization process. The TU shape is always square amd it may take a size from 32×32 down to 4×4 smaple. For an inter CU, the TU can be larger than CU, and it may cantain PU boundaries. However, the TU cannot cross PU boundaries for an intra CU.

The quadtree structure of multiple TUs within a CU is shown as Fig.2.5. For a CU with $2N \times 2N$ partitioning, a flag is used to decide whether the CU is split into four TUs. The best mode can be choose for a TU. The energy can be focused for larger size TU, and more details can be preserved for smaller size TU. This flexible structure can make the residual energy fully compressed, and improve the coding gain further.

2.1.2 Intra prediction

In H.265/HEVC, intra prediction of the luma component supports five PUs: 4×4, 8×8, 16×16, 32×32 and 64×64, and each PU has 35 prediction modes

$R_{0,0}$	$R_{1,0}$	$R_{2,0}$	…	$R_{N,0}$	$R_{N+1,0}$	$R_{N+2,0}$	…	$R_{2N,0}$
$R_{0,1}$	$P_{1,1}$	$P_{2,1}$	…	$P_{N,1}$				
$R_{0,2}$	$P_{1,2}$			…				
…	…							
$R_{0,N}$	$P_{1,N}$	…		$P_{N,N}$				
$R_{0,N+1}$								
$R_{0,N+2}$								
…								
$R_{0,2N}$								

Figure 2.6 The intra prediction template

which contain planar mode, DC mode and 33 angle modes. The prediction template is shown as Fig.2.6, and $R_{x,y}$ and $P_{x,y}$ represent the reference pixels of neighboring PU and the prediction pixels of the current PU, respectively. It is noted that the bottom left pixels are used as reference pixels, which in some cases, can improve the coding efficiency significantly.

All HEVC intra prediction modes are defined by prediction mode number as follow: planar (mode number:0), DC (mode number:1) and angle modes(mode numbers:2-34). The prediction directions of the 33 angle mode is shown as Fig.2.7. The direction of the mode number 2-17 mean horizontal modes, and the direction of the mode number 18-34 mean vertical modes [14].

- Planar mode

Planar mode corresponds to plane mode in H.264/AVC, and it adapts to the pixels smooth areas. The prediction pixel value $P_{x,y}$ is generated by the average of the prediction horizontal and vertical values. This method can make the change of prediction pixel smooth, and improve the video subjective quality.

- DC mode

DC mode is suitable for large flat areas. The current prediction value is generated by the average of the left and above reference pixels. That is the average value of $R_{0,1}$,... $R_{0,N}$, $R_{1,0}$,...,$R_{N,0}$ in Fig 2.6.

- Angle modes

There are eight different prediction directions in H.264/AVC. However, in order

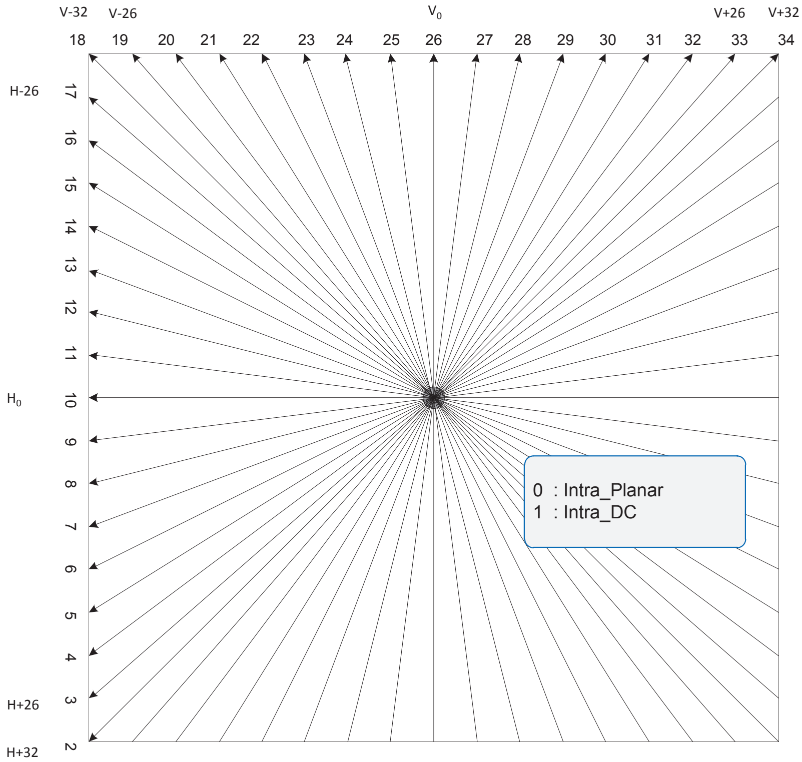

Figure 2.7 The prediction directions of the 33 angle mode

to adapt to the different texture of the video content, H.265/HEVC specifies 33 angle prediction modes in Fig.2.7. V_0 and H_0 represent the vertical and horizontal directions, and the prediction directions of other mode can be seen as a deviation in vertical or horizontal directions.

2.1.3 Inter prediction

In H.265/HEVC, the prediction block (PB) is the basic process unit in inter prediction, and the prediction unit contains the prediction informations. The motion compensation principle is that the reference blocks are used to predict the current block information. The displacement between the reference block and the current block is called motion vector (MV) and the difference between them is named motion distortion. The MV and motion distortion are used to determine the best prediction mode based rate-distortion (R-D) model [15].

Similar to H.264/AVC, B-frame or P-frame prediction is used for motion compensation, and in the final standard, the bi-prediction is used to achieve a trade-off between encoding efficiency and encoding complexity. Furthermore, it needs to ac-

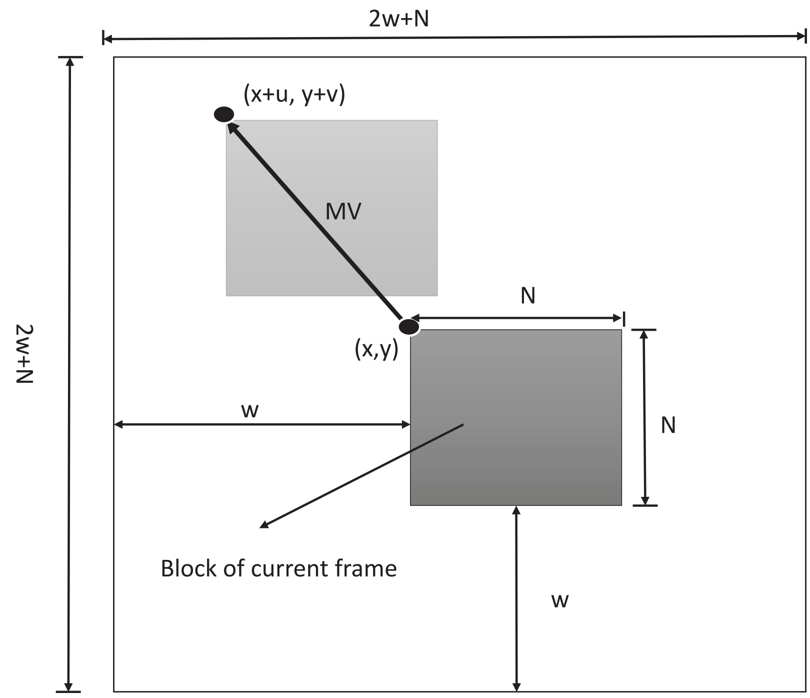

Figure 2.8 Block-based motion estimation

cess memory constantly for bi-prediction, which s considered to be the main factors of computational complexity, especially for hardware design.

- motion estimation

Motion estimation is the mainly technology in video compression. The most motion search is based on the block matching algorithm (BMA) [16]. BMAs assume that all the pixels within a block have the same motion activity, and the motion motion is based on the motion vector of rectangular blocks. Thus, BMA are more suitable for a simple hardware realization because of their regularity and simplicity. In the process of BMA, one is required to find a MB in the reference frame within a given search area, that is most similar to the MB in the current frame. Due to a given search range a window like structure is formed in the reference frame, which is known as the search range(SR). For the search range [-w, +w] and the block size N×N, the relationship between the current block and the SR is shown in Fig.2.8.

The matching criterion of the BMA has a direct impact on coding efficiency and computational complexity. Many matching criteria have been proposed in previous work: mean squared error (MSE), sum of absolute differences (SAD). For the SAD matching criterion, it needs some simple computational steps, and is suitable for VLSI design. The formular of SAD for a given location (u, v) is that:

$$SAD(u,v) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} |f(x,y) - f(x+u, y+v)| \tag{2.1}$$

where -w $\geq$ u, v $\leq$ +w; f(x, y) and f(x+u, y+v) are the current block pixels value and the reference block pixels value. In motion search precessing, the smallest SAD is selected as the best matching reference block, and the displacement between them is the current MV. In general, motion estimation algorithms are broadly classified into two categories: full search algorithm and fast search algorithms.

In full search algorithm, all search candidates within the search window are evaluated, and the search candidate with the smallest SAD is selected as the best matched search candidate. The final MV is obtained from the location of the best matched search candidate. Although full search yields optimum results, it requires a huge amount of computation.

In order to reduce the complexity of motion search, some fast searach algorithms are proposed in previous works. The fast search algorithm can reduce the complexity, however, the mainly disadvantage of fast search algorithm is that it may be trapped in some local minima and thereby produce suboptimal results. Fast search algorithms can be broadly classified into three categories: (1) reduction in the number of search candidates (2) exploiting different matching criteria instead of the classical SAD , and (3) predictive search based on their characteristics.

(1) reduction in the number of search candidates: the method is based on assumptions that the distortion monotonically decreases as the search candidate approaches the optimal one. Even if all the search candidates are not matched, the optimal search candidate can be obtained by following the search candidate with the smallest distortion. There are many algorithms available in previous works, such as 3-step search, new-3-step search, 4-step search, diamond search and so on.

(2) exploiting different matching criteria: The different matching criteria can impact the coding efficiency and computational complexity. Some criterias are proposed in previous works: MSE, SAD and so on.

(3) predictive search: The main disadvtage of fast search methods is that they

may trapped into a local minimum. In order to slove this problem, predictive search is used and combined with other fast search methods. Owing to the correlation between the current block and neighboring blocks, the motion information of neighboring blocks in the spatial or temporal space is used to serve as the initial search candidate of fast search algorithms instead of the original point.

- motin vector prediction (MVP)

In H.265/HEVC, the motion vector of horizontal component (x) and vertical component (y) differential is encoded, respectively. The two components motion vector difference (MVD) are defined as follow:

$$MVD_x = \Delta x - MVP_x \tag{2.2}$$

$$MVD_y = \Delta y - MVP_y \tag{2.3}$$

The motion vector of the current block has correlation with the motion vector of neighboring blocks in the spatial or temporal space. Thus, the MVP is generated by the neighboring blocks motion information. In H.265/HEVC, the advanced motion vector prediction (AMVP) technology is used to generate the motion candidate list, and the MVP candidate generating method is that: Firstly, two left spatial candidates are selected, Otherwise, the top spatial candidates are checked. Secondly, one temporal candidate is checked. When the selected candidate index is not more than 2, MV(0,0) candidate is added.

2.1.4 Transform and quantization

- Discrete cosine transform

HEVC specifies two-dimensional transforms of various sizes 4×4, 8×8, 16×16, and 32×32 that are finte precision approximations to the discrete cosine transform (DCT). Multiple transform sizes improve compression performance, but also increase the implementation complexity [17].

The N transform coefficients v_i of an N-point 1D DCT applied to the input samples u_i can be expressed as

$$v_i = \sum_{j=0}^{N-1} u_j c_{ij} \tag{2.4}$$

where i=0,...N-1. Elements c_{ij} of the DCT transform matrix C are defined as

$$c_{ij} = \frac{P}{\sqrt{N}} cos[\frac{\Pi}{N}(j + \frac{1}{2})i] \tag{2.5}$$

where i,j=0,...N-1 and where P is equal to 1 and $\sqrt{2}$ for i=0 and $i > 0$, repectively. The basis vector c_i of the DCT are defined as $c_i = [c_{i0}, ...c_{i(N-1)}]^T$ where i=0,..N-1.

There are several properties for DCT that are useful both for compression efficiency. (1). It is desirable for compression efficiency by achieving transform coefficients that are uncorrelated. (2). It provides good energy compaction which is also desirable for compression efficiency. (3). It is desirable for simplifying the quantization and de-quantization process. (4). It is useful to reduce implementation costs as the same multipliers can be reused for various transform sizes. (5). It is useful to reduce the number of arithmetic operations. (6). It can be utilized to implement fast algorithm.

For slowly changing grey value of pixels piece, after DCT most of the energy is concentrated in the upper left corner of the low frequency coefficient. On the contrary, if the pixel texture block contains more details information, more energy distributes in the high frequency area. In fact, most images contain more low frequency components. Using the characteristics that the human eye is not sensitive to high frequency detail image with relative, the low-frequency coefficients of high frequency energy can be handled subtly, and low energy of high frequency coefficients can be quantized roughly.

- Quantization

Quantization consists of division by a quantization step size (Qstep) and subsequent rounding while inverse quantization consists of multiplication by the quantization step size. In HEVC, quantization parameter (QP) is used to get Qstep, and QP can take 52 values from 0 to 51. The relationship between QP and Qstep is defined as follow:

$$Qstep(QP) = (2^{\frac{1}{6}})^{QP-4} \tag{2.6}$$

The integer DCT scaling operation need to complete at the same time in H.265/HEVC quantitative process. In order to avoid floating point arithmetic, quantizer formula (2.3) will enlarge to a certain extent both the numerator and denominator, then integer to retain the accuracy of operation.

In HEVC, the encoder can signal whether or not to use quantization matrices enabling freqency dependent scaling. Human visual system based quantizatin can achieve better quantily than frequency independent quantization. In HEVC, three options can be configured for the operation of the quantizer: flat quantization, default weighting matrix and custom weighting matrix.

The quantization step may need to be changed within a picture for rate control and perceptual quantization purposes. This is updated by a QP delta in the slice segment header. The applicable QP for a CU is derived from the QP applied in the previous CU in decoding order, and the dalta QP is transmitted in coding units with non-zero transform coefficients.

2.1.5 In-loop filters

HEVC includes two processing stages in the in-loop filter[18]: a deblocking filter and a sample adaptive offset (SAO) filter. The deblocking filter aims to reduce the visibility of blocking artefacts and is applied to sample located at block boundaries. The SAO filter arms to improve the accuracy of the reconstruction of the original signal amplitudes and is applied adaptively to all samples, by conditionally adding an offset value to each sample based on values in look-up tables defined by the encoder.

- deblocking filter

A deblocking filter process is performed for each CU in the same order as the decoding process. First vertical edges are filtered then horizontal edges are filetered. Filtering is applied to 8×8 block boundaries which are determined to be filtered, both luma and chroma componements.

The deblocking filter process has three stages: boundary decision, filter on/off decision and strong/weak filter decision. TU boundaries and PU boundaries are involved om the deblocking filter. In boundary decision stage, the boundary strength (Bs) is calculated to reflect how strong a filtering process may be needed for the boundary. A value of 2 for Bs indicates strong filtering, 1 means weak filtering and 0 means no deblocking filtering. The filter on/off decison is made using 4 lines grouped as a unit, to reduce computational complexity. If filtering is turned on, a decision is made between strong and weak filtering. The strong deblocking filter is applied to smooth flat areas.

- SAO filter

SAO is applied to the reconstructed signal after the deblocking filter by using offsets specified for each CTB by the encoder. The SAO reduces sample distortion by first classifying the samples in the region into multiple categories with as selected classifier and adding a specific offset to each sample depending on its category. The classifier index and the offsets for each region are signaled in the bitstream. SAO operation includes edge offset (OE) which uses edge properities for pixel classification in SAO type 1 to 4, and band offset (BO) which uses pixel intensity for pixel classification in SAO type 5.

2.1.6 Entropy coding

A single entropy coding scheme is used in all configurations of HEVC: context adaptive binary arithmtic coding (CABAC) [19]. Entropy coding is a lossless compression scheme that uses the statistical properties to compress data, and it is performed at the last stage of video encoding, after the video signal has been reduced to a series of syntax elements.

CABAC adopts efficient arithmetic coding technology, considers the related statistical properties video stream, and improves the coding efficiency significantly. Entropy coding processing has three stages: binarization, context modeling and binary arithmetic coding. In general, a binarization scheme defines a unique mapping of syntax element value to sequences of binary symbols, which can be interpreted in terms of a binary code tree. By decomposing each non-binary syntax element value into a sequence of bins, further processing of each bin value in CABAC depends on the associated mode decision. The probability models in CABAC are adaptive, which means that, for those high probability events on the coding performance, a delicate context model is set up, on the contrary, for the low probability events on coding performance, a simple context model is set up. For the syntax elements of binary, every Bin is processed with arithmetic coding according to the probability model parameters, and gets the final video stream. Binary arithmetic coding contains two kinds of encoding: regular coding mode and bypass coding mode. the regular mode uses the probability model of adaptive coding, and the bypass coding mode uses the form of equal probability coding.

2.2 Performances and problems

In the standardization of HEVC, the reference software, which is called HM has been developed as a common software platform for further improvement and study. The HM reference software is maintained at two sites: HHI and BBC. During the development of the HEVC specification, establishment of Common Test Conditions (CTC) provided a well-defined platform on which expriments for coding tool evaulations are performed.

For performance evaluation, CTC defines the four prediction structures: All Intra (AI), Random Access(RA), Low Delay P picture (LDP) and Low Delay B picture (LDB). For AI configuration, each picture is encoded as an I frame. For RA configuration, a hierarchical B structure is used. For LDP configuration, the first picture is encoded as an I frame and the subsequent frames are encoded as P frames. LDB is similar to the LDP configuration, the first picture is encoded as an I frame and the subsequent frames are encoded as B frames. Test sequences are defined accroding to the picture size and applications and they are classified into five classes (A to E). Class A to E are the set of test sequences with a picture size of 2560×1600, 1920×1080, 832×480, 416×240, and 1280×720 pixels, respectively.

In HEVC, R-D (Rate-Distortion) curve is used to evaluate the coding performance of a video codec, which is generated by plotting the encoded results, in terms of bit rate versus the video quality. In general, a high coding efficiency codec can achieve higher quality at lower bit rates. PSNR(Peak Signal to Noise Ratio) is used to evaluate the picture quality, and it is calculated for YCbCr component. In order to compare the coding efficiency, the average bit rate difference is referred to as BD-Rate (Bjontegaard Delta rate) and the average PSNR difference is referred to as BD-PSNR.

Compared with H.264/AVC, the compression efficiency of H.265/HEVC is over

H.264/AVC in both objective and subjective tests. Moreover, the bit rate reduction, based on objective evaluation of CTC test sequences, indicates all over performance improvement of about 50% over H.264/AVC. HEVC yields a substantial improvement in compression capability beyond that of H.264/AVC for video streaming applications, and the coding performance gains of HEVC over H.264/AVC generally increase with increasing video resolution up to at least 4K resolutions.

For the next generation of video coding, the features of parallel processing, high compression capability, and low computational complexity are very important. Firstly, with the advent of the era of multi-core, video coding with parallel processing is a trend. An embedded system for real time processing requirement is higher. Moreover, there are many computation for motion estimation, and it is very suitable for parallel processing. Therefore, in the future, there are both CPU and GPU in a SOC (System on a Chip). In HEVC, there are multiple parallel hierarchy, GOP level, frame level, slice level, tile level, CTB level, and CU level parallel. There are a bottleneck for CU parallel, and this work mainly focuses on CU level parallel. Secondly, from MPEG-2 to H. 264, it achieves 50% bitrate saving. From H. 264 to HEVC, it also achieves about 50% bitrate saving. Thus, high efficiency is the goal of video coding. Thirdly, it is demand of low computational complexity and low power consumption. With the improvement of coding efficiency, the computational complexity of video coding is higher. The coding time of inter prediction accounts for 69% of the total encoding time. Thus, reducing computational complexity is also the target of the next generation of video compression.

However, there are some challenges to achieve the above coding tools for the next generation of video coding. Firstly, there are data dependence between coding unit in HEVC. The dependence exists in intra and inter prediction. Secondly, the size of larger coding tree unit is biger, and there are more modes in intra prediction. The block partitioning methods are more diverse, and there are improvements for the transform unit.

In this work, we focus on the key technology for the next generation of efficient video coding. Firstly, the spatio-temporal prediction based approach is used to eliminate dependencies between CUs. Secondly, in order to improve the coding efficiency, an efficient motion vector prediction selection algorithm is presented. Thirdly, a fast CU size decision algorithm is presented to reduce the redundant computing for HEVC encoder side.

Chapter 3 Spatio-temporal prediction based algorithm for parallel improvement of HEVC

3.1 Introduction

High efficiency video coding (HEVC) is a successor to H.264/AVC video coding standard, which has a design target of 50% lower bitrate with largest coding unit (LCU) up to 64×64 [1]. To achieve this goal, there are some new tools in HEVC as shown in Table 3.1 which provides a comparison between some of the tools used in H.264 and HEVC.

Table 3.1 Comparison of tools in H.264 and HEVC.

Tools	H.264	HEVC
Coding unit size	8×8 to 16×16	8×8 to 64×64
Prediction unit size	4×4 to 16×16	4×4 to 64×64
Transform size	4×4 and 8×8	4×4 to 32×32
Intra prediction directions	Up to 9	Up to 35
Interpolation	1/4 sample(8-tap)	1/4 sample(6-tap)
Motion prediction	–	AMVP
Entropy coding	CABAC,CAVLC	CABAC
Loop filter	DF	DF, SAO

HEVC divides the picture into coding tree units(CTUs). The width and height of LCU are signaled in a sequence parameter set [6]. An LCU is further divided into four coding units (CU) in a quad-tree structure. And prediction unit (PU) is a region, defined by partitioning the CU. Each inter coded PU has a set of motion parameters including motion vector (MV), picture index, reference picture list. For each PU, there are three modes: inter, skip and merge. In the partitioning unit, the partition modes are used to define the prediction unit (PU) for inter-coded CU. Partition modes include two square partition modes, two symmetric motion partition modes and four asymmetric motion partition modes. Usually, the motion vector prediction candidate is selected, which minimized the cost J_{cost}.

Motion estimation (ME) is a well-established technique that has been employed in several video coding standards including MPEG-2, MPEG-4, and H.264/AVC. It is one of the most critical tools in video encoding which consumes more than 50% coding complexity. Many previous ME algorithms and architectures available for lower resolution video are not efficient enough to support HD or UHD videos, thus it is possible to optimize an existing ME algorithm or architecture to achieve a high efficiency. HEVC provides some parallel processing tools to speed up the encoding process on the slice level. However, parallel processing in CU level is not supported.

Previous works are proposed to facilitate the parallel processing and enlarge the throughput [7],[8],[20],[21] and [23]. G. Tian et al. exploited temporal, spatial and transform-domain features to speed up the original quadtree-based prediction [7]. By using this scheme, motion estimation was performed for prediction blocks within a narrowed range. Ching Chi et al. present a novel approach called overlapped wavefront (OWF), which achieves higher performance and efficiency than tiles and wavefront parallel processing (WPP). However, parallel processing will induce much coding efficiency loss [8]. X.T. Jiang et al. exploited a multistep composite method for integer motion estimation and eliminating redundancies for fractional motion estimation [20]. M.E. Sinangil et al. presented hardware cost vs. coding efficiency comparison for 11 different motion estimation configurations [21]. M. E. Sinangil el al. developed a hardware-aware search algorithm for HEVC motion estimation [22]. It can reduce the memory bandwidth and save search cycles because of target pixels sharing. G. Sanchez et al. presented a spread and iterative search and the low density and iterative search motion estimation algorithm [23], and it included a lot of data dependencies.

In order to process parallel motion estimation, the CU data dependence within a LCU have to be removed. In HEVC, advanced motion vector prediction (AMVP) is the search center, which is pre-defined on the basis of the surrounding available MVs before the motion estimation process. Owing to the correlations of MVs, the traditional method is to search every point with each block size one by one. As shown in Fig.3.1, within a CU, current block prediction depends on the motion information of neighboring blocks (A, B and C). After generating neighboring blocks prediction, the current block can process motion estimation. The dependency of blocks makes

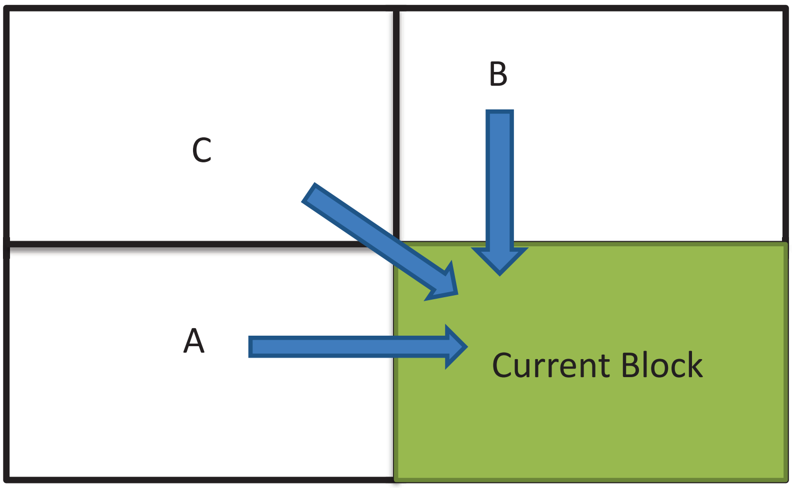

Figure 3.1 Dependence between encoding blocks

ME hard to be processed in parallel. So the neighbor blocks motion information could be generated concomitantly. However, there is no available parallel coding tool on a CU level due to its inherent coding order restriction.

Previous works are presented to improve the motion vector prediction [25]-[29]. But most of proposals bring coding efficiency loss. J. Park et al. proposed an improved method on median motion vectors (MVs), in which MVs with a different reference index or a different reference list [25]. K. Kazui et al. presented the improvement of simplified motion vector prediction, which extends the original SMVP scheme so that the possibility to obtain the maximum number of motion vector prediction candidates for a target motion vector improves with small increase of computational complexity [26]. J. Zhang et al. exploited the potential parallelism in HEVC mode decision (MD) process and proposed a highly parallel MD method, which worked in a motion estimation region (MER)[27]. Q. Yu et al. provided a parallel advanced motion vector prediction (AMVP) candidate list (AMVPCL) construction solution, which mode the motion vectors for all PUs within current CU available at the same time to improve the parallelizing degree of the Motion Estimation process at CTU level [28]. C.G. Yan et.al proposed a parallel framework to decouple motion estimation for different partitioning on many-core processors based on local parallel method (LPM) [29]. This approach achieves more than 2 times speedup compared with HM baseline, however, the bitrate is increased.

In this paper, a parallel spatial prediction algorithm is presented to get an enhanced AMVP. Then a temporal prediction candidate algorithm is proposed. At last a joint spatio-temporal prediction algorithm is presented to determine the motion information. This algorithm can easily estimate the spatial and temporal candidates to achieve parallel determining of AMVP. Furthermore, it is vital to note that the presented proposal is an improvement of HEVC, which make it necessary to modify the decoding tools on the decoder side.

3.2 Overview of the Inter Prediction in HEVC

3.2.1 Motion Vector Prediction

Motion vector prediction exploits spatio-temporal correlation of motion vector with neighboring PUs, which is used for explicit transmission of motion parameters. As shown in Fig.3.2, in inter prediction, there are two types motion vector candidates as AMVP candidates including spatial motion candidates and temporal motion candidates:

1. Left candidate(A_1), left bottom candidate(A_0);
2. Above candidate(B_1), above right candidate(B_0) and above left candidate(B_2);
3. Temporal candidates(C_0 and C_1).

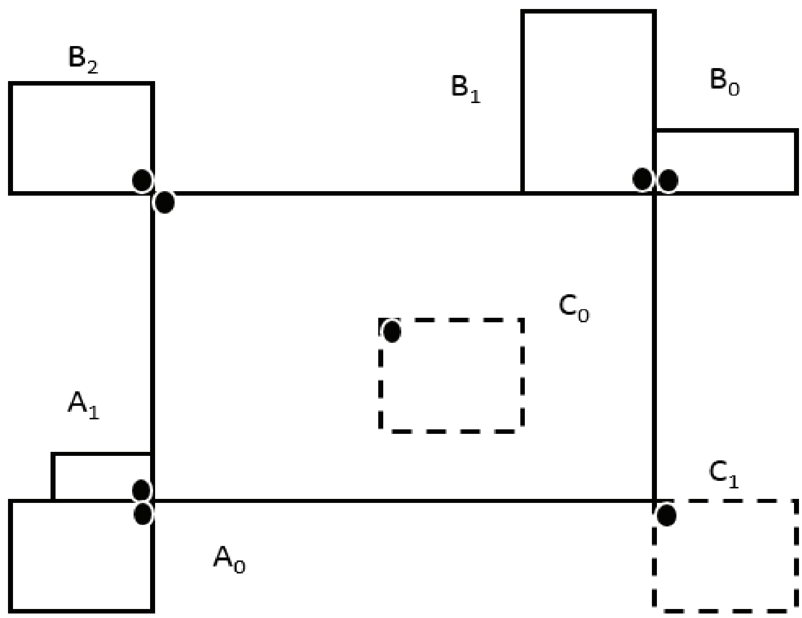

Figure 3.2 AMVP candidates

The process for motion vector prediction candidate is as follow: Firstly, for spatial candidate list $\{A_0, A_1, B_0, B_1, B_2\}$, two motion vector candidates are selected from it. When the left side of current PU is available, the left side of current PU is checked. The order is $\{A_0 \rightarrow A_1 \rightarrow$scaled $A_0 \rightarrow$scaled $A_1\}$. Otherwise, the above side of current PU is checked. The order is $\{B_0 \rightarrow B_1 \rightarrow B_2 \rightarrow$scaled $B_0 \rightarrow$scaled $B_1 \rightarrow$scaled $B_2\}$. Secondly, one motion vector candidate is derived from the temporal candidate list $\{C_0, C_1\}$. After that, the duplicated MV candidates should be deleted. If the generated candidate index is not more than 2, zero MV candidate will be add to the generated candidates. Otherwise the candidate indexes are more than two, the MV candidate will be removed, whose index is larger than 1.

Although motion vector of the neighbor block can improve the prediction accuracy of search center. This dependency of spatial motion vector is obstacle to construct the motion candidate in parallel. To enable spatial motion candidate producing in parallel, the motion vector outside current PU is useful to derive the spatial motion candidate. After getting the motion candidate, the motion estimation can be processed in parallel.

3.2.2 Reference Frame Management

There are three kinds of reference frames in HEVC: intra prediction frame (I-frame), predicted frame (P-frame) and bidirectional predictive frame (B-frame). The reference frame lists are divided into two groups (L0 and L1). Multiple reference frames are used to increase the coding performance. In order to improve the ability of resist errors, HEVC uses the reference picture set (RPS) technology, which is to transfer the frame change of decoded picture buffer (DPB) in the stream. So the current frame DPB state are not dependent on the previous frame DPB state. HEVC supports multiple reference frames. As shown in Fig.3.3, in HEVC reference software, there are 2 or 4 reference frame, which induce significant high complexity. In HEVC software model, the block matching technique is used to do motion search and select the best prediction and reference index.

Temporal motion information between neighbor block is important reference for current block. In order to generating the temporal motion vector in parallel, the motion vector of time $t-1$ block can be used to get the motion vector of

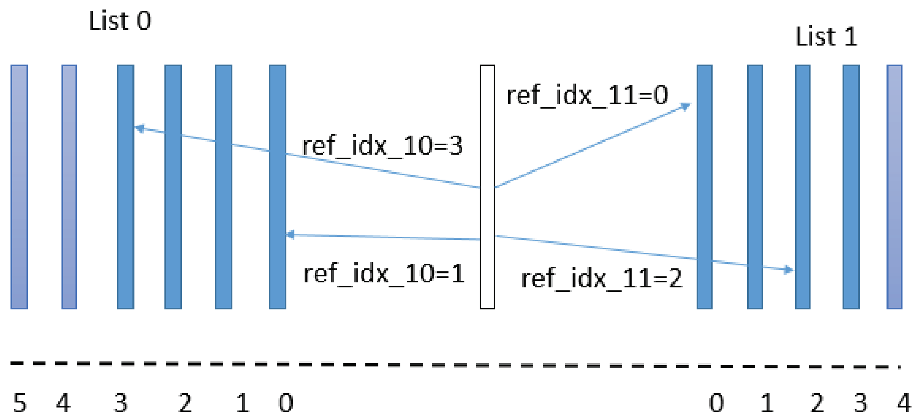

Figure 3.3 Reference frames

current block. Temporal motion vector could be generated in parallel. In the motion estimation stage, the motion search can process in parallel, which can decrease largely the encoding time.

3.3 The Proposed Coding Tools for Motion Candidate

In general, HEVC uses parallel some tools to improve parallel motion estimation (PME). HEVC encoder appoint a parallel motion estimation region (PMER). Within a PMER, some prediction block's motion information are available, but some prediction block's motion information are unavailable. There are three level parallel: CU-group level, CU level and sub-CU level. However, there is data dependence between coding units. The CU-group level parallel is to enable full parallel motion candidate list derivation for all CUs within a PMER. For the current block, the available spatial neighboring blocks can be utilized for motion candidate list, and the unavailable of spatial neighboring block can be removed or replaced by the available motion candidate. The temporal motion vector can be used to achieve the CU level parallel. Another CU level parallel is to derive motion candidate for a CU only once, and the same motion candidate is shared by all the prediction blocks inside the CU. Sub-CU level parallel is to remove the dependency of the second, third and fourth PUs. However, most of proposals are not to consider the encoding block consistency. So the consistency and parallel should be balanced in a suitable method.

3.3.1 Parallel Spatial Prediction Algorithm

Previous experiments prove that the bigger block can be encoded in whole, when the encoding block content is smooth [30]. There is consistency between motion vectors for motion estimation process. Furthermore, the encoding efficiency can be improved and complexity can be reduced. In contrary, when there are plenty details in encoding block, the current block should be divided into some sub-blocks. Because the motion information of different small block is different. When the encoding block content is smooth, motion information of the neighboring block is useful for the current PUs. Within a PMER, the unavailable motion candidates can be replaced by the outside motion candidate.

Within a PMER, the size is $2N \times 2N$. The spatial candidate list is $\{A_0, A_1, B_0, B_1, B_2\}$. Let mvx_{A_1}, mvx_{A_0}, mvx_{B_1}, mvx_{B_0} and mvx_{B_2} be the motion vector horizontal of spatial candidate list, respectively. In this work, a novel parameter which is used to evaluate the reliability of the candidate MVs is defined as

$$R_S = r_1 + r_2 \tag{3.1}$$

where r_1 is the absolute motion vector difference among A_0, A_1 and B_2, and r_2 is the absolute motion vector difference among B_0, B_1 and B_2. r_1 and r_2 are defined as

$$r_1 = |mvx_{A_1} - mvx_{B_2}| + |mvx_{A_0} - mvx_{B_2}| \tag{3.2}$$

$$r_2 = |mvx_{B_1} - mvx_{B_2}| + |mvx_{B_0} - mvx_{B_2}| \tag{3.3}$$

The algorithm for generating spatial motion candidate in parallel is described as follow:

Setp 1. Set the threshold value $TH1$. The parallel motion estimation region size is 64×64, and the CU depth is 0.

Setp 2. When the partition is $2N \times 2N$ in inter prediction mode, the spatial candidate list of current PU is ψ =$\{A_0, A_1, B_0, B_1, B_2\}_{2N \times 2N}$.

Setp 3. If the reliability parameter $R_S < TH1$, others partition ($2N \times N$, $2N \times N$, $N \times N$, $2N \times nU$, $2N \times nD$, $nL \times 2N$ and $nR \times 2N$) mode motion information $\{MV_1, MV_2 ...\}$ is subset of ψ. The motion vector prediction candidate is selected with the minimum cost J_{cost}, compared with the current cost.

Setp 4. Otherwise, the encoding block is divided into sub-blocks and the depth is incremented 1, repeat step 2,3.

Table 3.2 The number of AMVP candidates.

CU Size	**HM**	**Proposed**
64×64	13	1
32×32	13×4	4
16×16	13×16	16
8×8	5×64	64
Total	593	85

The presented method not only removes the PU dependency, but also reduces the number of motion candidates for spatial prediction. Table 3.2 shows that, for a PMER, the number of motion vector is 1 in the ideal case, which is 1.6% of HM case. In the worst case the number of motion vector is 85, which is just 14% of HM case. As the above analysis, the proposed algorithm provides the friendly parallel spatial prediction MVs. And threshold $TH1$ can balance rate-distortion and encoding time.

Fig.3.4 shows an example of generating the spatial candidate list. For the a 64×64 PMER, in inter mode with partition $2N \times 2N$, CU depth is zero. When the R_S is less than $TH1$, the spatial candidate list of others partition mode can be replaced by the partition $2N \times 2N$ motion information. Otherwise, the PMER is divided into sub-blocks (CU0, CU1, CU2 and CU3) and the depth is incremented 1. For CU0 in inter mode with partition $2N \times 2N$, when R_S is less than $TH1$, the spatial candidate list of the other partition mode can be replaced by the partition motion information of $2N \times 2N$. Otherwise, CU0 is divided into sub-blocks and the depth is incremented until the depth is equal to 3. Repeat the same process for CU1, CU2 and CU3, recursively.

3.3.2 Parallel Temporal Prediction Algorithm

In HEVC reference software, reference frames and co-located reference are used to generate the temporal motion vector($mvLxCol$) . As shown in Fig.3.5, when $mvL0$($mvL1$) or $mvCol$ is defined as a long-term frame, $mvLxCol$ is equal to $mvCol$. Otherwise, $mvLxCol$ is obtained by the scaling vector $mvCol$.

In general, HEVC has temporal motion vector prediction (TMVP) candidate list $\{ C_0, C_1\}$. When the TMVP is available, it can generate the precision predic-

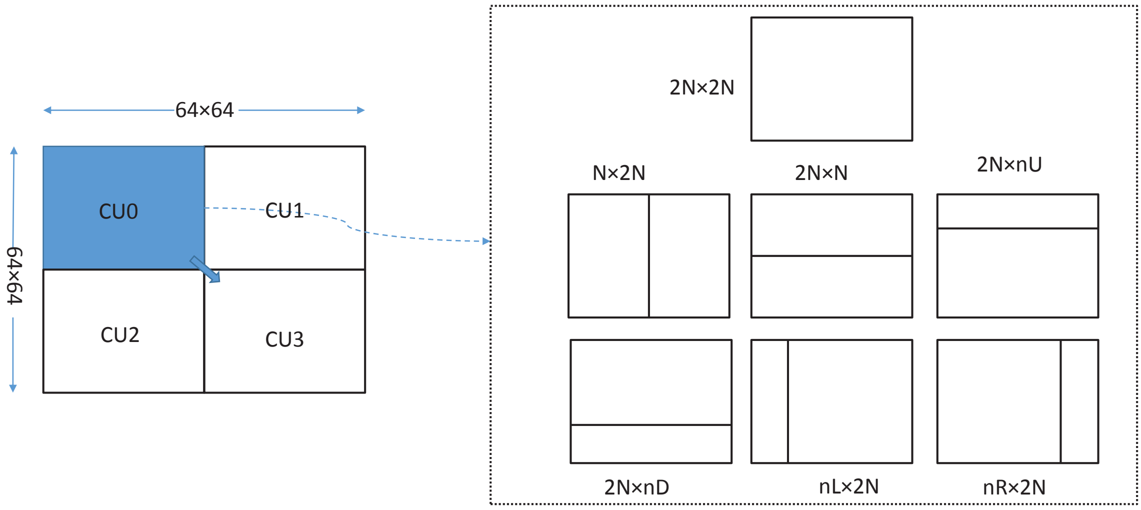

Figure 3.4 PU spatial prediction candidate determining

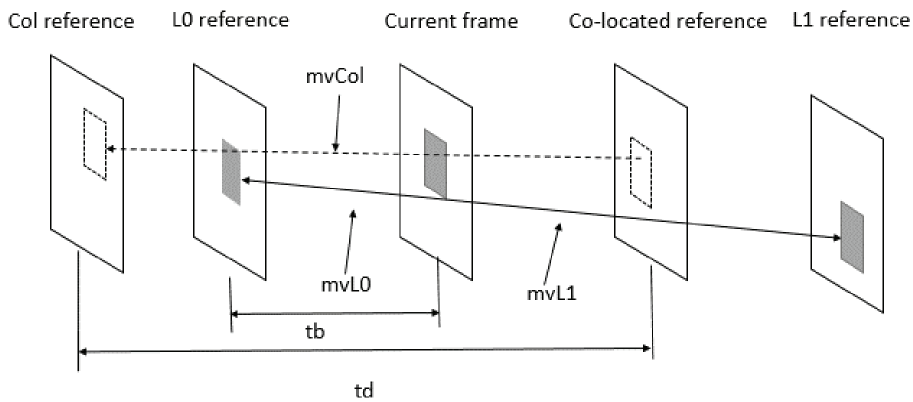

Figure 3.5 Temporal prediction

tion center. In this paper, simulation experiment prove the available rate of TMVP with different encoder configuration using 5 class test sequence. In HEVC software, there are four kinds of configuration for reference frame : intra-only(IO), low-delay P (LDP), low-delay B (LDB) and random access (RA). Experiment illustrates that the available rate TMVP in the LDP condition is more than 60%. In this paper, a temporal candidate determining algorithm is proposed dedicated to the RA condition.

In almost typical video sequences, not only the encoding block is smooth, but also the object motion is horizontal. In this case, the temporal motion vector of current encoding block in time t is similar with the motion vector of homologous block in time $t-1$. Let mv_t and mv_{t-1} indicate the motion vector in time t and $t-1$. When the object motion is horizontal, mv_t and mv_{t-1} are important information to achieve the temporal motion prediction. Using the motion vector consistency, it is easy to estimate the search center. Then it is possible to realize the motion estimation in parallel.

Within a PMER, the size of current block is 64×64. Let $\{mv(t)_x, mv(t)_y\}$ be the temporal motion candidate of current block in time t, and $\{mv(t-1)_x, mv(t-1)_y\}$ be the temporal motion candidate of past block in time (t-1). Another reliablity parameter R_T are defined as

$$R_T = |mv(t)_x - mv(t-1)_x| + |mv(t)_y - mv(t-1)_y| \tag{3.4}$$

Through the above analysis, the temporal motion candidate is derived according to the motion information of the corresponding block in time as follow:

Setp 1. Set the threshold value $TH2$. The parallel motion estimation region size is 64×64, and the CU depth is 0.

Step 2. When the partition is $2N \times 2N$ in inter prediction mode, the temporal candidate list of current PU is $\Omega = \{MV_t\}_{2N\times 2N}$.

Step 3. If the reliability parameter $R_T < TH2$, others partition mode motion information share the motion information Ω.

Step 4. Otherwise, the depth is incremented by 1, and repeat step 2,3 until the CU depth is equal to 3.

This above algorithm reduce the complexity of encoding block. It is noted that the encoder determine the temporal candidate only once for all different prediction mode within a PMER.

Fig.3.6 demonstrates an example of derivation temporal motion candidate. Within a large coding unit, the size is 64×64. When the partition of current PU is $2N \times 2N$ which inter mode, the temporal motion candidate is T. If the reliability parameter R_T is less than the $TH2$, The others partition of PU shared temporal candidate T. That is, for the other partitions ($2N \times N$, $2N \times N$, $N \times N$, $2N \times nU$, $2N \times nD$, $nL \times 2N$ and $nR \times 2N$), block 0 and 1 have the same temporal candidate T. Otherwise, the LCU is divided into sub-blocks (block 0, 1, 2 and 3), which means CU splits from 64×64 to 8×8 and the depth is incremented by 1. In the every depth,

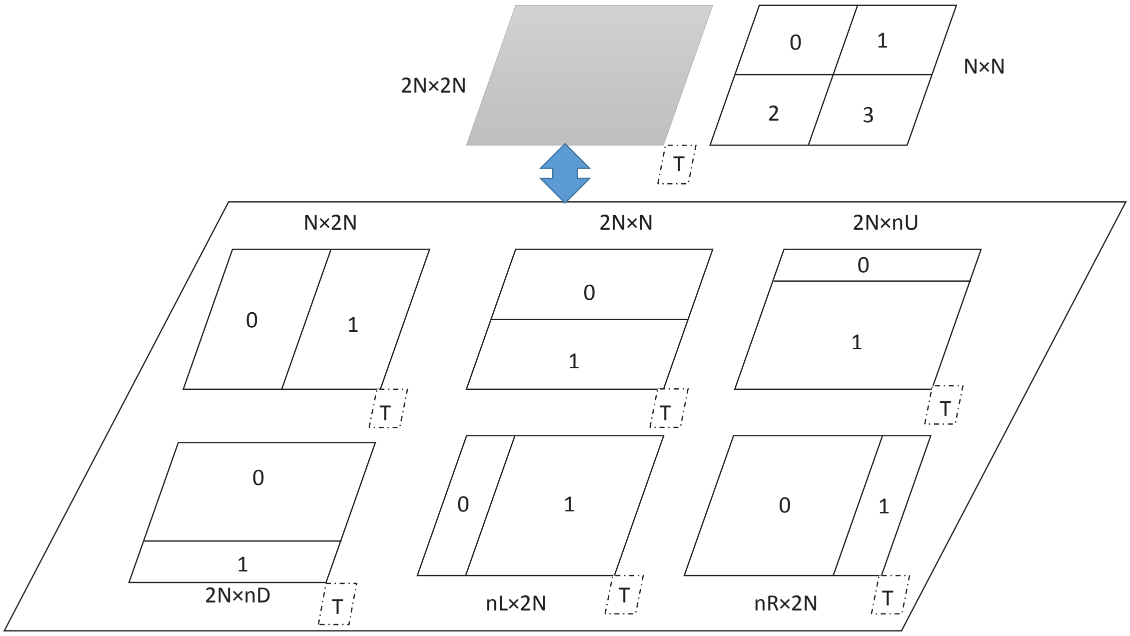

Figure 3.6 PU temporal prediction candidates determining

the different PU partition share the temporal candidate T of the partition $2N \times 2N$ when it meets the certain condition.

3.3.3 Joint Parallel Spatio-Temporal Prediction

In general, the image content is with abundant texture. Owing to the different distance between neighboring block, spatial and temporal motion information are important to adaptively predict the motion candidate of current block. Some images have compactness relation in spatial space, and some images have closeness relation in temporal space. To avoid further bitrate loss, a limitation on both spatial and temporal motion candidates are introduced. However, from the viewpoint of parallelism performance, the limitation on both spatial and temporal motion candidates will decrease the parallelism performance. Therefore, the condition of the joint spatial-temporal prediction algorithm is selected by a tradeoff between the bitrate loss and the parallelism performance.

Another parameter is introduced to indicates the parallel space (SP), which is defined as

$$SP = R_S - \mu R_T \tag{3.5}$$

where μ ($\mu > 0$) is the factor between R_S and R_T. $TH1$ and $TH2$ are the threshold value of spatial and temporal motion difference (R_S, R_T). μ is defined as

$$\mu = \frac{TH1}{TH2} \tag{3.6}$$

Obviously, When $SP >= 0$, $R_S >= \mu R_T$, the parallel space is temporal space. But when $SP < 0$, $R_S < \mu R_T$, the parallel space is spatial space.

Based on the parallel spatial prediction and parallel temporal prediction, the joint adaptive motion information candidate prediction algorithm for parallel improvement of HEVC is summarized as follows:

Step 1: Set the threshold value $TH1$ and $TH2$, and start AMVP for PUs with one CU in the inter mode.

Step 2: PU partition is $2N \times 2N$, the AMVP candidate list is $\{MV_1, MV_2, ...T\}$. Compute R_S, R_T and r using Eq (1), (4) and (5).

Step 3: When $R_S < TH1$, $R_T < TH2$ and $R_S >= \mu R_T$, the AMVP candidates of other partitions is generated using temporal motion candidate of $2N \times 2N$: $\{T\}_{2N \times 2N}$. When $R_S < TH1$, $R_T < TH2$ and $R_S < \mu R_T$, the AMVP candidates of other partitions is generated using spatial motion candidates of $2N \times 2N$: $\{MV_1, MV_2, ...\}_{2N \times 2N}$.

Step 4: Otherwise, CU depth is incremented by 1, and repeat step 2,3 until the CU depth is equal to 3.

3.3.4 Parallelism Performance Analysis

In this work, the dependence between prediction units has been removed. With the proposed methods, all PUs within parallel motion estimation region (PMER) can generate motion vector candidates parallelism. After that, ME can process in parallel. As the Fig.3.7 is shown, in a parallel motion estimation region (PMER), the AMVP candidates of each PU can be generated in parallel. After the dependency of the four neighboring block is removed, all PUs within one CU can process motion estimation in parallel.

There are many concurrency platforms such as single instruction multiple data (SIMD), Multi-thread platform, Multi-core processor platform, Compute Unified Device Architecture (CUDA) and so on. Motion estimation include two modules,

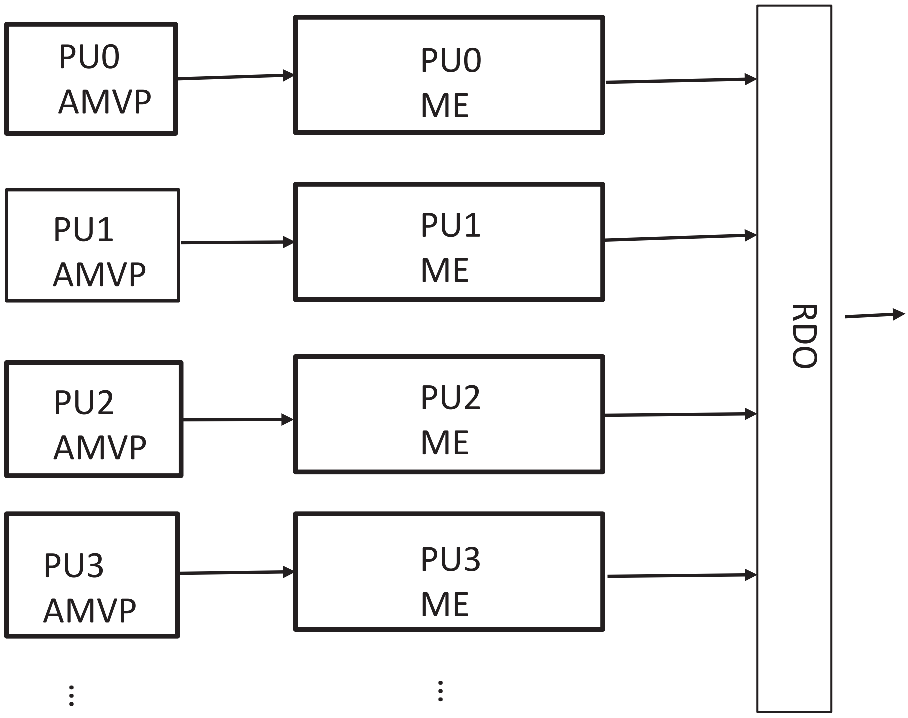

Figure 3.7 Parallel encoding in encoder side

integer motion estimation and fractional motion estimation. Integer motion estimation usually performs a coarse search over the whole search region, which has high parallelism requirement. Using above tools, it is able to implement parallel motion estimation. It is worth noting that this work is an extension of HEVC, and it is beyond HEVC. Furthermore, the two threshold are fixed values. Thus, they need not to transfer to the decoder for the extension of HEVC.

In the available MVs, temporal MVs have very high co-relation with the current PMV. Therefore, our proposed algorithm consider not only the spatial but also the temporal MVs as the prediction candidates to improve the prediction performance. When limited spatial MVs are available, the temporal MVs play an important roles to improve the accuracy of the alternative MVs. However, storing of all mv_{t-1} is considered as an additional workload. This workload is not a critical issue for software implementation. Because for a typical software implementation of a HEVC, comparing to the frame memory, the memory capacity for mv_{t-1} is small enough.

3.4 Experiment and Results

The proposed algorithms is tested and verified based on HEVC software model HM12.0. The experiment platform is a typical PC with INTEL Core i7-4770 3.4GHz x64 CPU and 8.00G of RAM. The reference test sequences are listed as Table.3.3. The configuration of random access (RA) main profile is used. For each sequence, quantization parameter (QP) value of 22, 27, 32 and 37 are specified in our experiments.

On the basis of our simulation results for different resolutions and different sequences, we find that the distribution of the reliability parameters (R_S and R_T) are an approximate gaussian distribution, which is shown in Fig.3.8. From Fig.3.8, it is also clear that optimal thresholds of TH1 and TH2 will significantly influence the parallelism and bitrate. If the threshold is too big, the bitrate will increase. On the contrary, a too small threshold will cause almost no gain of parallelism. Many simulations are performed on difference resolution and characters. Two typical results are shown in Fig.3.9 and Fig.3.10 for TH1 and TH2, respectively. For TH1, the bitrate shows almost the same coding performance when TH1 value is close to 21. For TH2, the difference of the bitrate is small when the range of TH1 value about 8. When the TH1 and TH2 values are gradually increased, the bitrate extends to rise sharply. In this paper, the threshold values of TH1 and TH2 are set to 21 and 8 considering the tradeoff between parallelism degree and bitrate increase.

The performance of the proposed algorithm is evaluated by Bjontegarrd Delta Bit Rate (BDBR) and BDPSNR according to the reference [31], and the average time saving (TS) is defined as

$$TS(\%) = \frac{1}{4}\sum_{i=1}^{i=4} \frac{T_{HM}(QP_i) - T_{pro}(QP_i)}{T_{HM}(QP_i)} \times 100\% \tag{3.7}$$

where $T_{HM}(QP_i)$ and $T_{pro}(QP_i)$ are the encoding time by using the HEVC reference software and the proposed method with different QP.

3.4.1 Spatial prediction

Table.3.4 shows the simulation result of proposed spatial prediction algorithm, which reveals the bitrate increase and PSNR loss. It is noted that the average BD-rate is increased 2.74% that is because the MV is not the nearest prediction center.

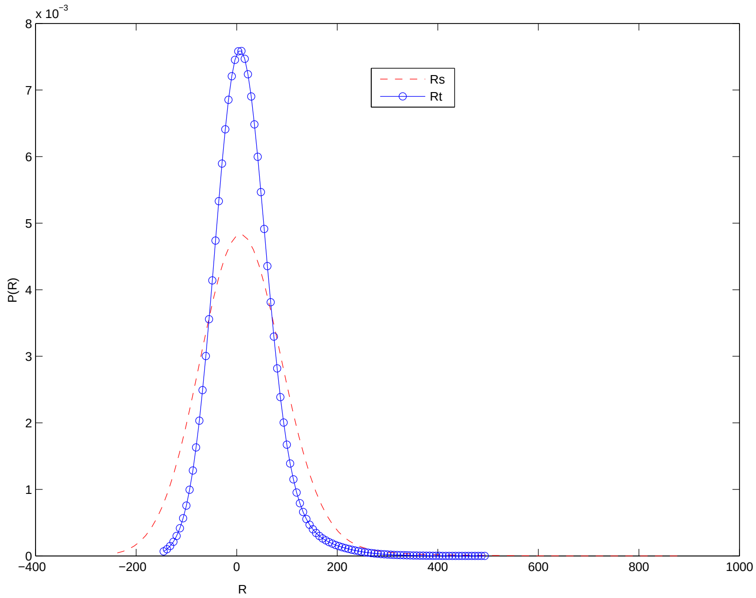

Figure 3.8 The distributions of R_S and R_T in Vidyo4 sequence when QP=32.

The reduced of average PSNR quality is only 0.003% under the same BD-rate.

The R-D curves of SlideShow sequence is shown as Fig.3.11 with different QP value, respectively. It is also noticed that the encoding performance of proposed method is same as HM12.0 test model. From the low bitrate to high bitrate test sequences, R-D curves are close to HM reference software.

3.4.2 Temporal prediction

Table.3.5 shown the result of presented temporal prediction algorithm, compared with the HM12.0 test model. It is noticed that the average BD-rate is increased 2.10%, and the average PSNR is raised only 0.13%. We notes that the bitrate loss is smaller than the front spatial approach.

Furthermore, the R-D curves of test sequences is shown as Fig.3.12. The simulation result shows that the coding performance is almost the same as HM12.0 test model. The PSNR of both the low bitrate and high bitrate test sequences are almost unchanged.

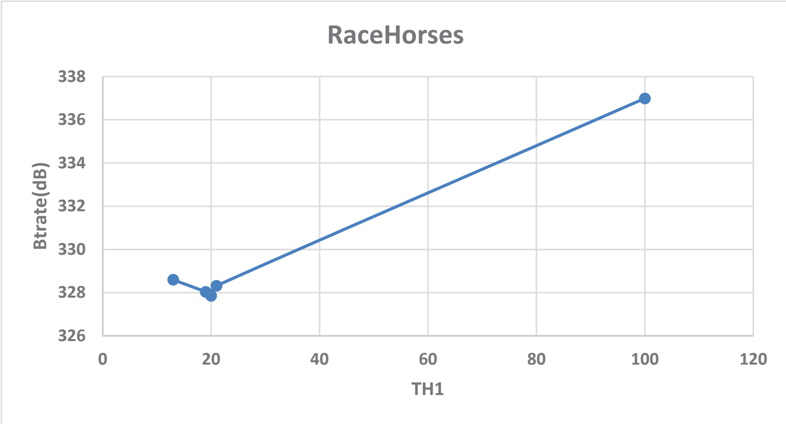

Figure 3.9 The bitrate curve with differnet TH1 values in RaceHorses sequence when QP=32 and framelength=64.

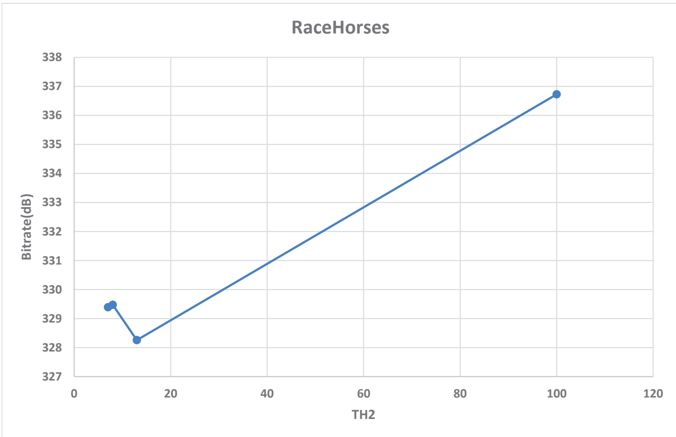

Figure 3.10 The bitrate curve with differnet TH2 values in RaceHorses sequence when QP=32 and framelength=64.

3.4.3 Spatio-Temporal Prediction

This section demonstrates the result of the joint spatio-temporal prediction algorithm in parallel for HEVC as Table.3.6. When the parallel space is spatial area, the other partition mode prediction candidates are replaced by the outside of MER's spatial candidate. When the parallel space is temporal area, the temporal prediction candidate of the current block is replaced by the prediction information of the previous block. This approach combines with the motion homogeneity of spatial and temporal prediction. It brings the smallest bitrate loss among spatial method, temporal method and spatio-temporal method, and the R-D curves are almost un-

Table 3.3 Test sequences.

Resolution	Sequence	Frame Length
416 × 240	**BasketballPass**	64
	BQSquare	64
	BlowingBubbles	64
	RaceHorses	64
832 × 480	**Flowervase**	64
	Keiba	64
	Mobisode	64
1280 × 720	**KristenAndSara**	64
	SlideEditing	64
	SlideShow	64
1920 × 1080	**Tennis**	64
	BasketballDrive	64
	Kimonol	64

Table 3.4 The performance comparison of spatial prediction with HM12.0.

Resolution	Sequence	BDBR	BDPSNR
416 × 240	**BasketballPass**	2.5	-0.119
	BQSquare	3.61	-0.159
	BlowingBubbles	2.08	-0.081
	RaceHorses	1.75	-0.084
832 × 480	**Flowervase**	2.78	-0.098
	Keiba	7.23	-0.274
	Mobisode	2.06	-0.048
1280 × 720	**KristenAndSara**	0.85	-0.034
	SlideEditing	7.02	1.075
	SlideShow	1.15	-0.102
1920 × 1080	**Tennis**	0.60	-0.018
	BasketballDrive	3.07	-0.065
	Kimonol	0.95	-0.031
Average		2.74	-0.003

changed.

The proposed algorithm achieves that the average BD-rate is increased only 0.01%. For the high resolution applications, compared with C.G.Yan's work[14], the average BD-rate loss of sequences are 0.1% and 0.5%, respectively, while the proposed method almost has no bitrate loss. Furthermore, the R-D curves of SlideShow is shown as Fig.3.13. The simulation result shows that the coding performance is almost the same as HM12.0 test model.

The simulation results in Table 3.6 show that the proposed method slightly increase on average the encoding complexity on the multicore platform. The average encoding time increased only 0.74%, compared with HEVC reference software. For

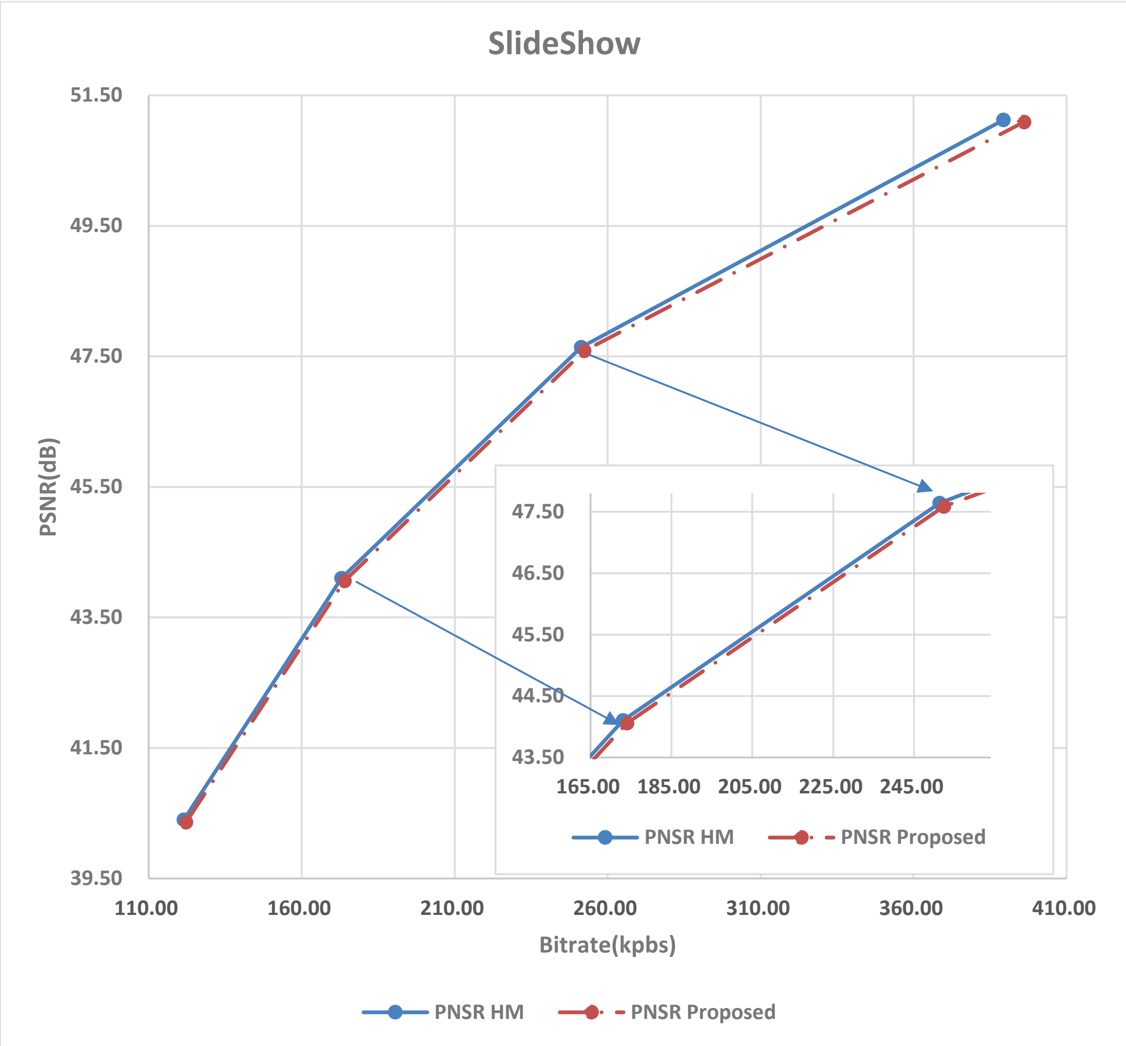

Figure 3.11 R-D curve for spatial prediction

the low resolution (416 × 240), the average encoding time reduced 3.55%, and the benefit of the complexity reduction achieved from the early termination of PMV prediction are much more than the demerit that induced by the TZSearch. Although for some sequences in high resolution (Flowervase, Keiba), the encoding time increased by the error of PMV estimation, the average encoding time increased only 2.64%, which is negligible for total encoding time.

It is noticed that the BDBR of the spatio-temporal prediction algorithm is lower than the spatial prediction algorithm and temporal prediction algorithm. The reason is that, in the joint spatial-temporal prediction algorithm, not only the parallelism

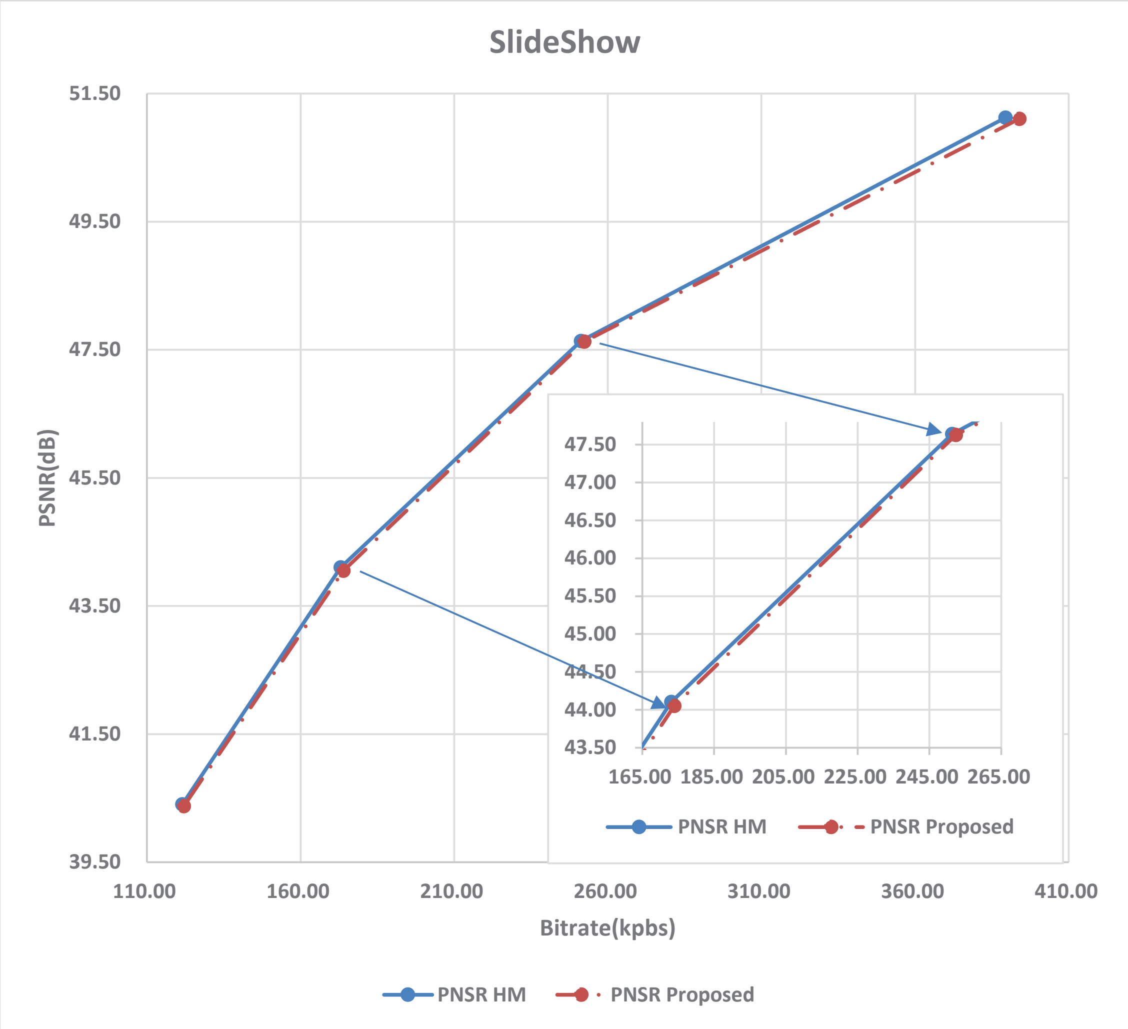

Figure 3.12 R-D curve for tempormal prediction

performance but also the bitrate loss are considered.

Through the above analysis of the results, it is clear that the BD-rate has a little change. Also the compression of the proposed method does not change compared with HEVC test model. Using the presented algorithm, the complexity is reduced to determine the motion information. Moreover, the parallelism of the proposal is improved largely, which is an increasingly important element of modern processing architectures.

Table 3.5 The performance comparison of temporal prediction with HM12.0.

Resolution	Sequence	BDBR	BDPSNR
416 × 240	**BasketballPass**	1.76	-0.082
	BQSquare	2.01	-0.089
	BlowingBubbles	1.49	-0.058
	RaceHorses	2.04	-0.097
832 × 480	**Flowervase**	1.48	-0.052
	Keiba	6.72	-0.262
	Mobisode	1.20	-0.029
1280 × 720	**KristenAndSara**	0.88	-0.028
	SlideEditing	5.42	-0.831
	SlideShow	0.85	-0.073
1920 × 1080	**Tennis**	0.78	-0.024
	BasketballDrive	1.99	-0.044
	Kimonol	0.65	-0.021
Average		2.10	-0.130

Table 3.6 The performance comparison of spatial-temporal prediction with previous work.

		Proposed Method			**[14]'s Method**	
Resolution	**Sequence**	**BDBR**	**BDPSNR**	**TS**	**BDBR** (8 × 8)	**BDBR** (16 × 16)
416 × 240	**BasketballPass**	-0.03	0.002	4.37	NA	NA
	BQSquare	0.07	-0.003	5.50	NA	NA
	BlowingBubbles	0.02	-0.001	2.56	NA	NA
	RaceHorses	0.02	-0.001	1.76	NA	NA
Low Average		0.02	-0.001	3.55	NA	NA
832 × 480	**Flowervase**	0.11	-0.003	-11.44	0.3	0.9
	Keiba	0.08	-0.003	-7.46	0.0	0.5
	Mobisode	-0.03	0.001	3.99	0.0	0.6
1280 × 720	**KristenAndSara**	0.10	-0.005	3.99	0.3	0.7
	SlideEditing	-0.32	0.050	-5.59	-0.4	0.1
	SlideShow	0.22	-0.021	-4.22	0.1	0.5
1920 × 1080	**Tennis**	-0.10	0.003	-0.82	0.1	0.5
	BasketballDrive	-0.01	0.000	6.64	0.1	0.6
	Kimonol	-0.02	0.000	0.16	0.2	0.5
High Average		0.00	0.002	-2.64	0.1	0.5
Total Average		0.01	0.001	-0.74	NA	NA

3.5 Summary of this chapter

This paper proposed a spatio-temporal prediction algorithm to improve the parallelism of motion estimation in HEVC. Firstly, using the character of the encoding block, a spatial prediction algorithm is presented. Secondly, a temporal prediction algorithm is proposed to remove the dependency of encoding block, which make to determine the temporal motion vector only once within a PMER. At last, a joint

Figure 3.13 R-D curve for spatio-tempormal prediction

spatial and temporal prediction algorithm is proposed to improve the encoding complexity in parallel. And the experiment results shown that the BD and PSNR was similar to the origin test model. In the future, focusing on the motion estimation, a parallel algorithm based on parallel platform will be developed.

Chapter 4 Coding efficiency improvement of HEVC by adaptive selection of prediction motion vector

4.1 Introduction

High efficiency video coding (HEVC) is the latest video coding standard which is released in 2013 [1]. In HEVC, the maximal size of basic coding unit (CU) is 64×64, and the size of search range is 64×64 for motion estimation (ME). Furthermore, the advanced motion vector prediction (AMVP) technique is used to determinate the search center. Compared with H.264/AVC, HEVC achieves about 50% bitrate saving while the computation complexity is increased significantly.

In the motion estimation process of HEVC, a high accuracy prediction motion vector (PMV) is introduced to define an accurate search center to save coding bit. The PMV is selected from a motion vector candidate list which consists of one motion vector from neighboring units on the left of the current coding unit, one motion vector from above neighboring units, and the motion vector of the spatially the same position in the previous encoded frame. And the motion vector in the list with minimum cost is selected as the final PMV. However, the fixed pattern of the PMV decision process without consideration of the reliability of the surrounding motion vectors makes it has lower estimation accuracy. Moreover, the strong correlation between the coding units make it difficult to do parallel coding.

To enhance the coding efficiency, some coding tools are used to improve the coding performance. However, the encoding complexity has significantly increased. There are many previous works which are proposed to improve the encoding efficiency and reduce the encoding complexity. These work can be divided into two groups: (1). Improve the encoding efficiency [24], [32]-[36]. (2). Reduce the encoding complexity [37]-[42].

A framework for better motion vector (MV) and skip mode is proposed, and the predictors are selected by a rate-distortion criterion in [32]. In these methods, the simple spatial median is selected using the spatial and the temporal redundancies in the MV fields. The reference picture management is modeled as an optimization problem and approximated its optimal solution in [33]. These methods can achieve average 7%-9% bitrate saving. However, the encoding complexity has highly increased. J.L. Lin et.al [24] propose motion vector coding techniques, including a priority-based derivation algorithm for spatial and temporal motion candidates, a surrounding-based candidate list, and a parallel derivation of the candidate list. This method can improve the encoding efficiency for the MV coding in HEVC, which achieve on average 3.1 % bitrate saving. W.H. Peng et.al [34] introduce an inter-frame prediction technique that combines two MVs derived from template and block machine for overlapped block motion compensation. Moreover, multi-hypothesis prediction and motion merge are used to achieve the trade-off between the encoding efficiency and complexity, and it achieve on average about 2% bitrate saving. The encoding efficiency improvement of HEVC is proposed in [35], and the asymmetric motion partitioning (AMP) is used for inter prediction. A new selection algorithm is proposed to improve the accuracy of prediction motion vector in [36]. Furthermore, the an adaptive motion search range algorithm is designed. However, the bitrate saving of this method is only 0.16% on average.

C.C. Lou et.al [37] proposed an adaptive motion search range (SR) prediction algorithm. In these work, a probability model between MV and prediction motion vector (PMV), and it can reduce the memory access bandwith significantly. Z. Shi et.al [38] propose a fast motion estimation algorithm named simulated annealing adaptive search (SAAS) to reduce the encoding computational, and MV correlation information is used in this proposed method. Z.P. Deng et.al [39] propose an iterative motion and disparity estimation algorithm, in these work, an adaptive search range adjustment is designed through the confidence measure of the constraint. Adaptive search range is determined based on the probability of motion vector differences (MVDs) [40], the proposed method can considerably reduce computation while conserving PSNR close to the HM reference software. A. Paul proposes an adaptive search window based on prediction error of the neighboring blocks [41], in this work, the threshold vales are chosen for different search range. T.K. Lee et.al [42] propose an efficient search range algorithm to reduce the computational complexity for HEVC by depth intensity mapping. The aforementioned methods are able to reduce computations while the coding efficiency is close to HM reference software.

In this work, an accurate prediction motion vector selection algorithm is proposed, and significant bitrate saving is achieved by increasing the prediction accuracy of PMV. Moreover, a adaptive search range (ASR) selection algorithm will help to reduce the motion estimation (ME) computational complexity which takes up most of encoding time in HEVC encoder side. The proposed method can achieve a better trade-off between the encoding efficiency and encoding complexity.

4.2 Motivation

4.2.1 Motion Vector Prediction in HEVC

Encoding MV for each CU can bring significantly bitrate saving, especially when small CU is chosen. Moreover, there are high correlation in MVs between current CU and nearby CU. The prediction motion vector (PMV) is formed based on the calculated MV and the difference motion vector (MVD). The PMV is depends on the CU size and the valid MV of nearby CU. In H.264/AVC, the method of forming the PMV depends on a component wise median of three spatially neighboring motion vectors. In HEVC, the approach of implicitly deriving the PMV was replaced by a technique known as AMVP, and the AMVP is significantly simplified to provide a good trade-off between coding efficiency and an implementation cost.

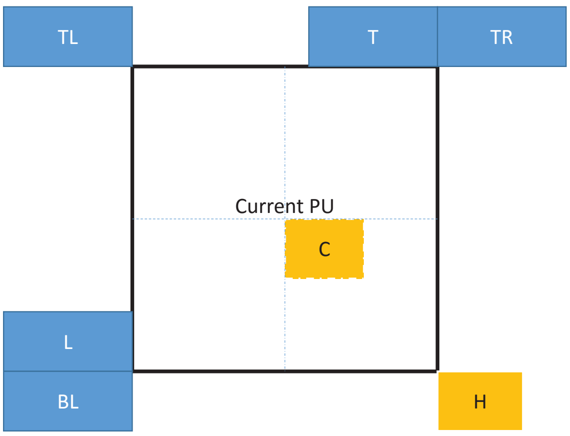

Figure 4.1 AMVP candidates

As shown in Fig.4.1, AMVP candidates of current PU include five spatial motion candidates: left candidate (L), left bottom candidate (BL), top candidate (T), top right candidate (TR), top left candidate (TL); and two temporal candidates (C and H). Firstly, two left spatial candidates are selected, Otherwise, the top spatial candidates are checked. Secondly, one temporal candidate is checked. When the selected candidate index is not more than 2, MV(0,0) candidate is added.

HEVC AMVP technique can improve the prediction accuracy of search center, compared with H.264/AVC. However, this PMV is not accuracy enough. The MVs of nearby CU can be used to improve the coding efficiency of current CU further.

4.2.2 Motion Search in HEVC

In video coding compression, motion compensation uses block-matching algorithm (BMA) to reduce the temporal redundancy between frames. The BMA algorithm is implemented to find a matching block from the reference frame. The matching criterion is evaluated by the cost function named sum of absolute difference (SAD) or Hadamard Transformed SAD (SATD) between the current block and reference block. In the sense of coding performance, the full search (FS) algorithm is used to search the best matching block in the all candidates within the given search range (SR). However, the computational complexity of FS algorithm is unacceptable. Many fast search algorithm is applied to reduce the computational complexity without the coding performance loss.

In HEVC, a fast search mechanism (FSM) called the test zonal (TZ) search is used to control the integer motion estimation (IME) complexity. The TZsearch algorithm is based on a fixed diamond search with refinement, and this algorithm consists of four distinctive stages. (1). motion vector prediction: median predictor, left predictor, up predictor and upper right predictor are employs to select the search center. (2). initial grid search: a diamond search or square search algorithm is used to find the search range (SR) (The default maximum search range sets to 64×64) (3). raster search: the raster search is a simple full search on the search window. (4). refinement: this step involves a greater accuracy of the MV obtained from the previous steps.

Although the TZsearch algorithm can reduce the computational complexity of ME to a certain degree, some computational redundancy can be reduced further.

4.3 Proposed PMV Selection Algorithm with ASR

In this section, the proposed high efficiency coding algorithm is described. Firstly, in order to find more accuracy prediction motion vector, the modified AMVP rule is used to improve the encoding efficiency. Then, after getting the better PMV, the adaptive search range adjustment algorithm is proposed to reduce the computational complexity in motion estimation process.

4.3.1 Proposed Prediction Motion Vector Selection Algorithm

The performance of ME is highly depend on the selection of MV candidates [25], [26], [43]. If the PMV is close to the calculated MV, the MVD between PMV and the calculated MV will be small, and the PMV is more accurate. However, in HEVC, a total of seven spatial and temporal MVs are added to candidate list to predict the MVP. There are two questions for the current AMVP mechinanism in HEVC. Firstly, the number of the reference MVs is low. Secondly, it is not adaptive for selecting the reference MVs. Therefore, it can not the accurate PMV using the current AMVP mechinanism. In order to improve the encoding efficiency further, the more reference MVs can be added to the candidate list. Owing to the motion consistency of spatial and temporal, the MVs surrounding the current coding unit is useful for determining the PMV. However, too much PMV candidates can cause too big amount of calculation. Thus, it is necessory to reduce the calculation redundancy.

With this point of view, in order to improve the encoding efficiency, the reference MVs of current coding unit can be used to search the accurate PMV. As Fig.4.2 shown, MV_{Temp} and MV_{Spat} are the MV of the temporal and spatial neighboring unit. MV_{TR}, MV_{TL}, and MV_{BL} indicate the MV candidates in the top right, top left, and bottom left of the current unit.

In the proposed algorithm, multiple approach are proposed to improve the accuracy of the PMV together with the consideration of the coding complexity reduction.

Since large coding unit is always selected in smooth frame as Fig.4.3. For the

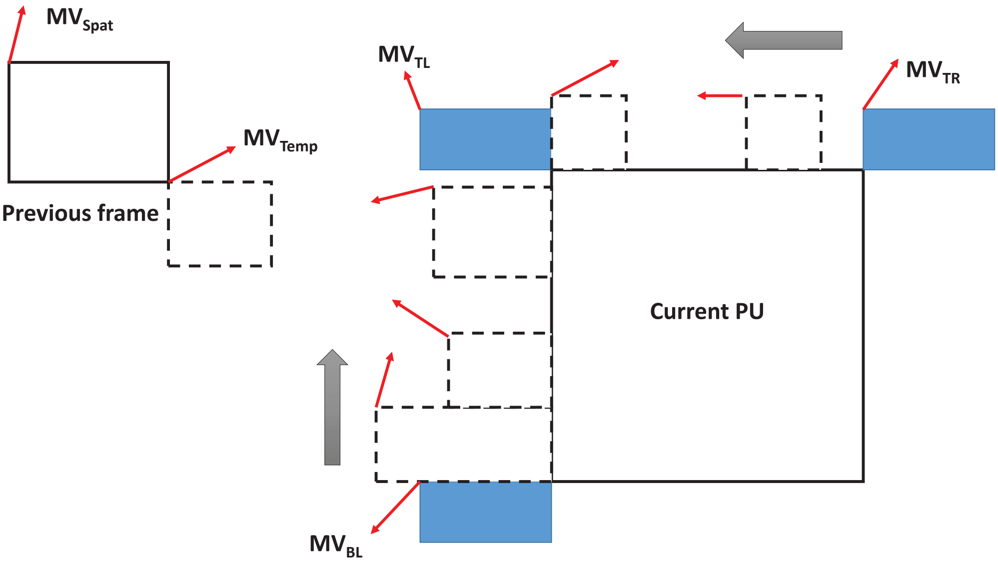

Figure 4.2 The reference MVs of current PU

coding unit of 64×64, it can get the accurate PMV using the reference MVs. Thus, only the first MV from the three candidates is selected as candidate PMV to early terminate AMVP mechanism. On the contrary, sometimes it can not accurately predict the PMV, and it need more reference MVs.

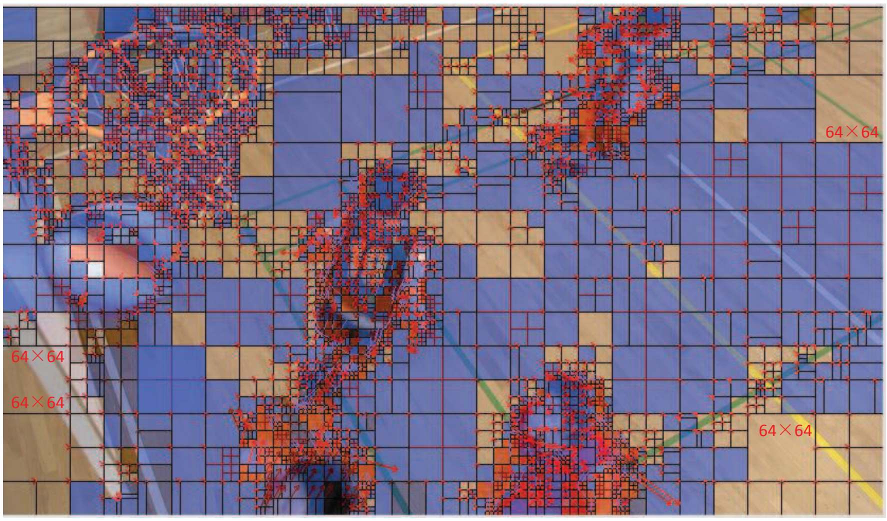

Figure 4.3 Large coding unit in smooth frame.

When the MV absolute difference (MVAD) between the spatial neighboring units and the temporal neighboring unit of the current coding unit is small, the texture of current coding unit tends to be not complex. Thus, the MVAD satisfies as

$$MVAD = |MV_{Spat} - MV_{Temp}| < 4 \tag{4.1}$$

the MV_{Spat} will be selected as the final PMV. These two approach can help to reduce the coding complexity.

Some other approach are proposed in the algorithm to improve the accuracy of the PMV candidates. As Jiang mentioned in [43], the reliability of the candidate MVs is evaluated by the reference MVs: MV_{TR}, MV_{TL}, and MV_{BL}. Thus the reference MVs satisfies as

$$MV_{TR} = MV_{TL} = MV_{BL} \tag{4.2}$$

The MV_{BL} will be selected as the final PMV. Furthermore, the absolute difference value (MVT) of the top left and top right, the absolute difference value (MVL) of the top left and bottom left are calculated as:

$$MVT = |MV_{TR} - MV_{TL}| \tag{4.3}$$

$$MVL = |MV_{TL} - MV_{BL}| \tag{4.4}$$

The two differences are compared to determine the MV in which direction is more reliable. When $MVT > MVL$, the PMV position is tended to the left of coding unit. Then MV_{TL} and MV_{BL} are selected as PMV candidates. Otherwise, the PMV position is tended to the top of coding unit, and MV_{RT} and MV_{TL} are selected as PMV candidates.

If all above early termination conditions are not satisfied, all the encoded MVs surrounding the current unit are added in the candidate list.

As aforementioned approach, the proposed prediction motion vector selection algorithm is shown as algorithm 1.

4.3.2 Adaptive Search Range Adjustment

The PMV selection algorithm can improve the encoding efficiency, however, it brings the encoding complexity increasing significantly. To reduce the encoding complexity of ME, after getting the accurate PMV, the adaptive search range (ASR) can be adjusted appropriately.

In general, a smaller search range is used for the case of the accurate search center. Moreover, a smaller search range is used for the higher depth of the coding unit. Thus, the search range for each coding unit in each iteration is adaptively

Algorithm 1: Proposed PMV selection algorithm.

Start inter prediction for a CU
if *CU size is* 64×64 **then**
  The first candidate is selected as best PMV
else if MV_{Temp} *exist* **then**
  if MV_{Spat} *exist,and* $|MV_{Spat} - MV_{Temp}| < 4$ **then**
    MV_{Spat} is selected as best PMV
else if MV_{TR},MV_{TL},MV_{BL} *exist* **then**
  if $MV_{TR} = MV_{TL} = MV_{BL}$ **then**
    MV_{BL} is selected as best PMV
  else if $|MV_{TR} - MV_{TL}| > |MV_{TL} - MV_{BL}|$ **then**
    MV_{TL} and MV_{BL} to candidate
  else
    Add MV_{TR} and MV_{TL} to candidate
else
  Add all neighboring MV to candidate
Reduce the redundant PMV candidate and compare the cost

adjusted by the depth of coding unit. This method can avoid being trapped in a local minimum with a reasonable search range.

With the different depth of coding unit, the size of the refinement SR can be adjusted as

$$SR = \begin{cases} 32,(\text{if } d_i < 2) \\ 16,(\text{if } d_i = 2) \\ 8,(\text{if } d_i = 3\ ) \end{cases} \tag{4.5}$$

where d_i and SR denote the current depth of coding unit, and the search range in the ME process, respectively. As Fig.4.4 shown, when the SR is set reasonably by different depth, the computational compexity of ME can be reduced with TZsearch search.

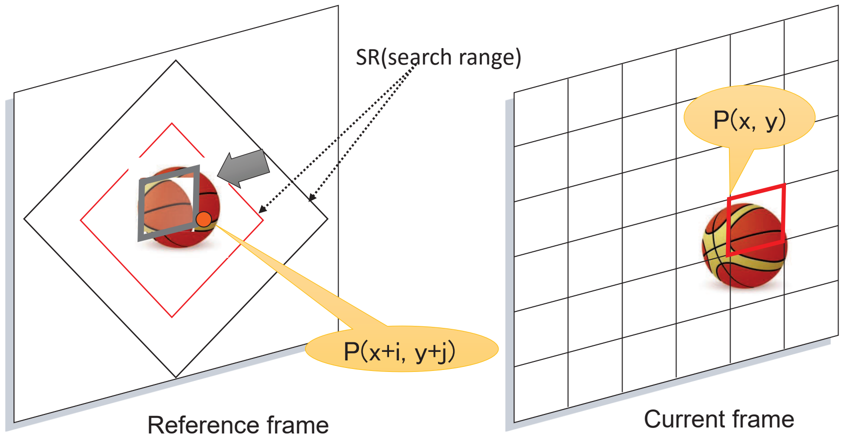

Figure 4.4 ASR for motion search

4.3.3 Overall Algorithm

Jointing the proposed prediction motion vector algorithm and adaptive search range adjustment, the flowchart of presented overall algorithm is shown as Fig.4.5. The overall algorithm can be divided into two distinctive steps as follow:

Step 1: Prediction Motion Vector selection.

The accurate PMV selecting algorithm is used to determine the search center. If the size coding unit is 64×64, the first valid MV candidate as the PMV. If the absolute difference between MV_{Spat} and MV_{Temp} is less than 4, the MV_{Spat} is be selected as the final PMV. If the absolute difference between MV_{TR} and MV_{TL} is more than the absolute difference between MV_{TL} and MV_{BL}, MV_{TL} and MV_{BL} are added to the candidates. Otherwise, MV_{TR} and MV_{TL} is added to the candidates. If the MV_{TR}, MV_{TL}, and MV_{BL} are invalid, all the encoded MVs surrounding the current unit are added to the candidates, and the redundant PMV candidate can be reduced according to the cost function.

Step 2: Adaptive Search Range Adjustment.

Search range refinement can reduce the search complexity in the ME process. It is performed when MV_{TR}, MV_{TL} and MV_{BL} are invalid. Search range is adjusted according to the different depth of coding unit.

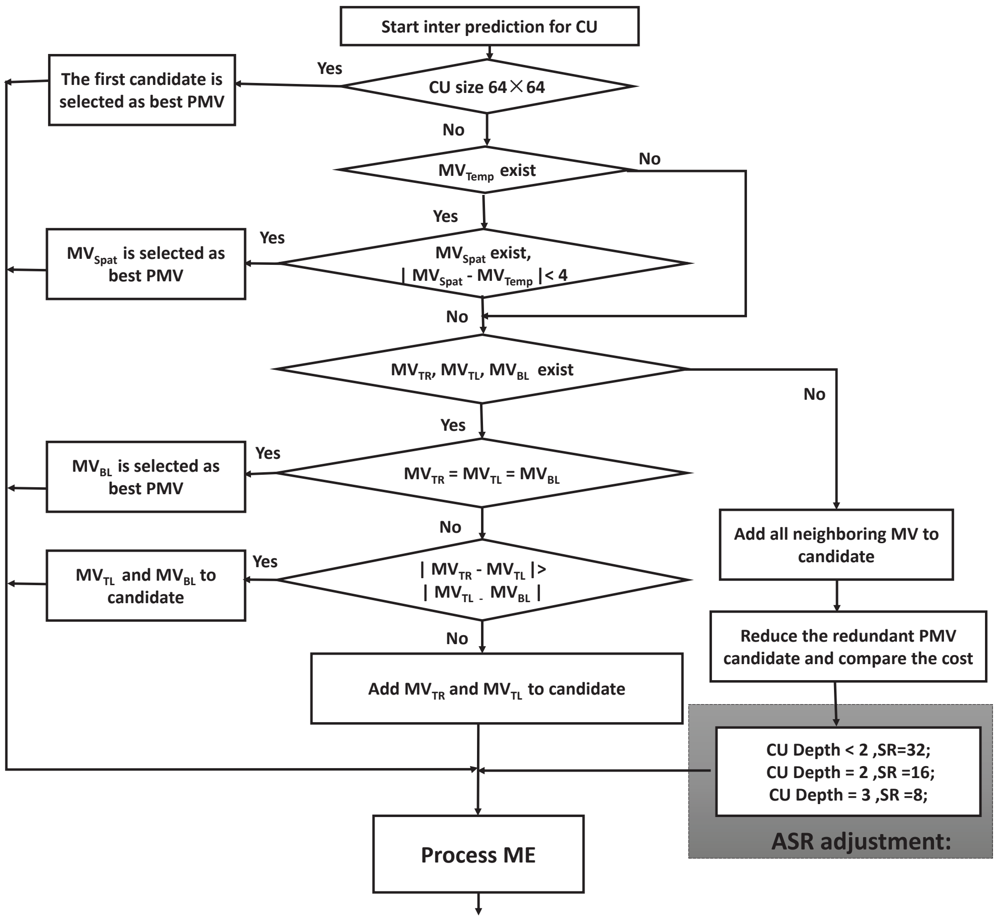

Figure 4.5 The flowchart of the proposed overall algorithm.

4.4 Experiment Results

The proposed algorithm is implemented and verified based on HEVC test model HM9.0. The quantization parameters (QP) are set to 22, 27, 32 and 37, respectively. The search strategy is TZsearch, and the search range (SR) is adjusted adaptively.

The performance of the proposed algorithm is evaluated by Bjontegarrd Delta Bit Rate (BDBR) and BDPSNR according to the reference [31]. The average complexity increased (CI) is calculated as

$$CI(\%) = \frac{1}{4}\sum_{i=1}^{4} \frac{T_{pro}(QP_i) - T_{HM}(QP_i)}{T_{HM}(QP_i)} \times 100\% \tag{4.6}$$

where $T_{HM}(QP_i)$ and $T_{pro}(QP_i)$ are the encoding time by using the HEVC reference software and the proposed method with different QP_i.

Table 4.1 The result of prediction motion vector selection algorithm.

		RA			LD		
Class	**Sequence**	**BD-BR**	**BD-PSNR**	**CI (%)**	**BD-BR**	**BD-PSNR**	**CI (%)**
1920 × 1080	**Kimono**	-6.24	0.214	159	-4.17	0.148	150
	ParkScene	-5.08	0.163	196	-5.10	0.166	112
	Cactus	-4.34	0.100	193	-4.57	0.108	99
	BasketballDrive	-6.82	0.134	202	-5.19	0.112	117
	BQTerrace	-2.90	0.063	92	-3.01	0.073	90
1280 × 720	**Vidyo1**	-5.30	0.184	101	-5.91	0.204	107
	Vidyo3	-5.45	0.175	98	-4.99	0.171	92
	Vidyo4	-6.27	0.183	104	-6.04	0.174	101
High Res.	**Average**	**-5.30**	**0.152**	**143**	**-4.87**	**0.145**	**109**
832 × 480	**BasketballDrill**	-7.77	0.327	94	-6.78	0.279	106
	BQMall	-4.70	0.208	96	-5.25	0.230	100
	PartyScene	-4.36	0.221	88	-4.47	0.226	97
	RaceHorses	-6.99	0.288	148	-5.78	0.246	163
416 × 240	**BasketballPass**	-6.50	0.331	96	-5.97	0.295	99
	BQSquare	-2.75	0.136	77	-4.40	0.210	83
	BlowingBubbles	-6.18	0.250	93	-6.24	0.250	96
	RaceHorses	-8.17	0.421	145	-7.06	0.368	173
Low Res.	**Average**	**-5.93**	**0.273**	**105**	**-5.74**	**0.263**	**115**
Average		-5.61	0.212	124	-5.31	0.204	112

4.4.1 Performance of PMV Selection Algorithm

The performance of the proposed PMV selection algorithm is shown as Table 4.1. It is shown that the average BDBR are -5.61%, and -5.31%, while the CI are 124% and 112% in the randaccess (RA) and lowdelay (LD) profile case, respectively. The results demonstrate that the proposed PMV selection algorithm can significantly improve the encoding efficiency. Furthermore, the high efficiency can be increased 8.17% for sequence RaceHorses.

Fig.4.6-Fig.4.7 show a typical example of R-D curve for Vidyo4 sequence in the RA and LD case. As it shown that the coding efficiency of the proposed PMV selection algorithm is better than HEVC reference software.

It is noted that this method brings the encoding complexity increasing with encoding efficiency improving. However, there are computational redundancy for motion search.

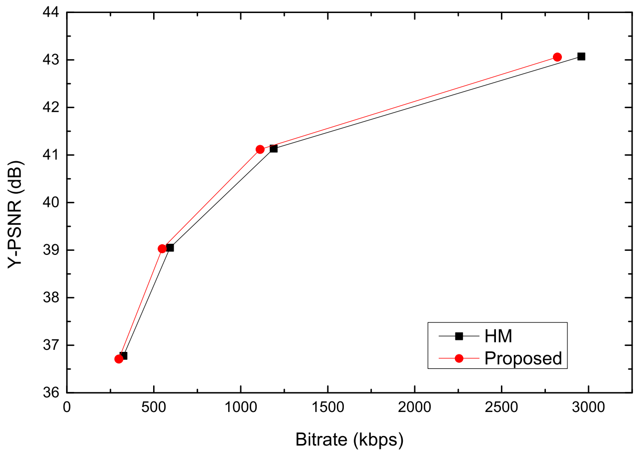

Figure 4.6 R-D curve for Vidyo4 with RA

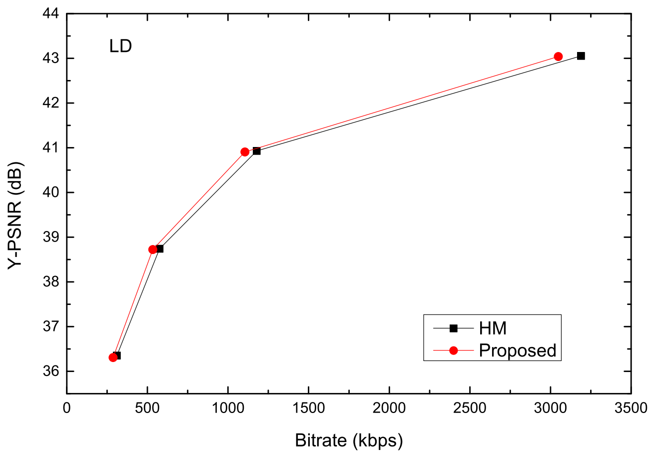

Figure 4.7 R-D curve for Vidyo4 with LD

4.4.2 Performance of Overall Algorithm

Table 4.2 shows the performance of overall algorithm. In RA case, the average BDBR is -5.22%, while the CI is 89%. In LD case, the average BDBR is -5.17%, while the CI is 91%. It is noted that the performance in low resolution is better than the performance in high resolution. Compare with the aforementioned PMV selection algorithm, the computational complexity can be reduced further by the search range adjustment, while the encoding efficiency can not be reduced.

In the RA and LD case, the R-D curves of the sequences, RaceHorses (Class D), Vidyo4 (Class E) are as shown in Fig.4.8-Fig.9. It is noted that, the encoding efficiency of the overall algorithm is better than HEVC reference software.

Table 4.2 The result of proposed combinative algorithm.

		RA			LD		
Class	**Sequence**	**BD-BR**	**BD-PSNR**	**CI (%)**	**BD-BR**	**BD-PSNR**	**CI (%)**
1920 × 1080	**Kimono**	-5.88	0.201	74	-4.10	0.146	107
	ParkScene	-4.91	0.157	96	-5.07	0.165	110
	Cactus	-3.99	0.093	92	-4.38	0.105	95
	BasketballDrive	-6.09	0.124	92	-4.80	0.100	89
	BQTerrace	-2.83	0.062	77	-2.93	0.071	91
1280 × 720	**vidyo1**	-4.89	0.172	94	-5.79	0.200	86
	Vidyo3	-5.16	0.169	89	-5.26	0.180	85
	Vidyo4	-6.16	0.179	94	-5.79	0.167	89
High Res.	**Average**	**-4.99**	**0.145**	**89**	**-4.77**	**0.142**	**94**
832 × 480	**BasketballDrill**	-6.76	0.284	79	-6.35	0.259	76
	BQMall	-4.45	0.197	84	-5.28	0.231	87
	PartyScene	-4.31	0.218	77	-4.43	0.225	80
	RaceHorses	-6.13	0.252	113	-5.25	0.223	118
416 × 240	**BasketballPass**	-6.29	0.319	86	-5.67	0.279	79
	BQSquare	-2.62	0.130	72	-4.45	0.212	70
	BlowingBubbles	-5.84	0.235	80	-6.16	0.248	74
	RaceHorses	-7.23	0.376	117	-6.97	0.365	112
Low Res.	**Average**	**-5.45**	**0.251**	**89**	**-5.57**	**0.255**	**87**
Average		-5.22	0.198	89	-5.17	0.199	91

Furthermore, in Table 4.2, the simulation results shown is the evaluation of the overall bitrate including the I slice. In order to evaluate the performance improvement of the overall algorithm for B slice, the evaluation results for B slice only are shown in Table 4.3. The proposed method can reduce about 10.06% of the bitrate on average for B slice only. In the best case, the encoding efficiency can improve 17.05% for Vidyo4 sequence. Therefore, the proposed method can significant improve the encoding efficiency of B slice.

Compared with previous work [24] and [34], the reference result is shown as Table 4.4. In both high resolution and low resolution, the coding efficiency of the proposed method is higher than Lin's and Peng's method. The benefit is from the accurate PMV according to the MVs surrounding the coding unit. Moreover, this proposed method can achieve a better tradeoff between the encoding efficiency and encoding complexity in HEVC.

It would be specially mentioned that this proposed method causes the encoding complexity increasing with the encoding efficiency raising. However, for the

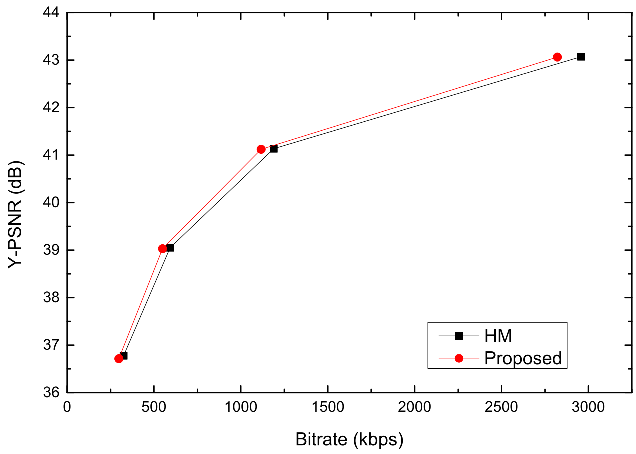

Figure 4.8 R-D curve for Vidyo4 with RA

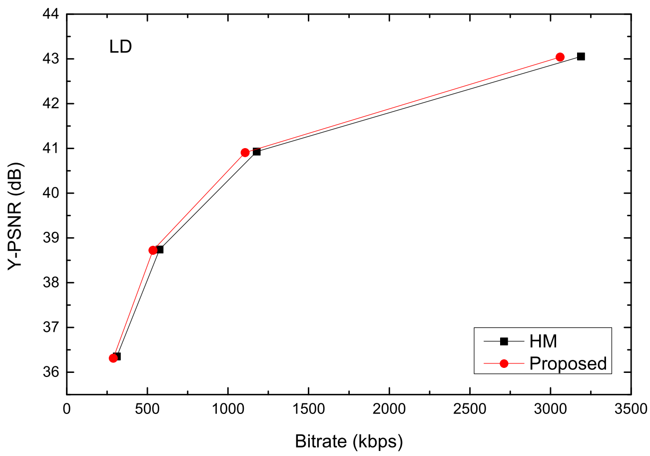

Figure 4.9 R-D curve for Vidyo4 with LD

application which does not care about the encoding time, care more about the coding efficiency, it is an efficient approach for improvemet of HEVC. Moreover, this increased encoding complexity can be reduced by the parallel computation in the feature.

4.5 Summary of this chapter

In this work, Firstly, an efficient motion vector prediction selection algorithm is presented. The MVs surrounding the coding unit are used to get the accurate PMV. After that, the search range is adjusted adaptively in motion estimation process. Finally, simulation results demonstrate that the proposed overall algorithm can improve the encoding efficiency 5.17%-5.22% on average, and the average complexity

Table 4.3 The result of B-Slice only.

Class	Sequence	BDBR(%)
1920 × 1080	**Kimono**	-7.66
	ParkScene	-9.77
	Cactus	-8.15
	BasketballDrive	-9.02
	BQTerrace	-7.63
1280 × 720	**vidyo1**	-13.57
	Vidyo3	-11.61
	Vidyo4	-17.05
High Resolution	**Average**	10.56
832 × 480	**BasketballDrill**	-11.56
	BQMall	-9.39
	PartyScene	-7.51
	RaceHorses	-8.02
416 × 240	**BasketballPass**	-11.41
	BQSquare	-6.74
	BlowingBubbles	-11.42
	RaceHorses	-10.44
Low Resolution	**Average**	-9.56
Average		-10.06

Table 4.4 The performance comparison with previous work.

		BDBR		
Config	**Method**	**High Res.**	**Low Res.**	**Average**
RA	**Proposed**	**-4.99**	**-5.45**	**-5.22**
	J.L Lin[24]	-2.30	-2.20	-2.23
	W.H. Peng[34]	-1.70	-1.80	-1.77
LD	**Proposed**	**-4.77**	**-5.57**	**-5.17**
	J.L Lin[24]	-3.90	-4.25	-4.20
	W.H. Peng[34]	-1.85	-2.15	-2.00

increased are 89%-91%.

Chapter 5 Fast CU size decision based on probabilistic graphical model in HEVC

5.1 Introduction

High efficiency video coding (HEVC) is the latest video coding standard that is recommended in 2013 [45]. It is a hybrid coding model, and it achieves 50% bitrate saving compared with H.264/AVC. HEVC divides the picture into coding tree unit (CTU). Then a CTU is further divided into four coding units (CU) in a quad-tree structure. Prediction unit (PU) is a smaller unit, defined by partitioning the CU. Each inter coded PU has a set of motion parameters including motion vector (MV), picture index, and reference picture list. For each PU, there are three available modes: inter, skip and merge mode. In the partitioning unit, the partition modes are used to define the prediction unit for inter-coded CU. Partitioning modes include two square partition modes, two symmetric motion partition modes and four asymmetric motion partition modes. Usually, the best motion vector is selected from certain motion vector prediction candidates which can minimize the rate-distortion (RD) cost.

In HEVC, the maximum CU size is 64×64, and each of this sub-CU can be divided into smaller sub-CUs recursively in the same quad-tree. As a result, the intra and inter prediction take the most of encoding time in the whole encoding process. The implementation cost is unacceptable for both software and hardware implementation, especially for those real-time applications. Therefore, on the premise of guaranteeing the coding performance, reducing the encoding computational compexity is a key factor for the success of HEVC, and the need of large scale HD video application.

However, CU size decision is vital process in HEVC, and the more researches focus on CU fast decision algorithm. The purpose of this kind of methods is to reduce the encoding complexity of CU size decision, on the premise of guaranteeing the coding quality. There are different between intra prediction and inter prediction. The intra CU size fast decision algorithm mainly targets the CU size decision of intra mode prediction, while inter CU size fast decision algorithm mainly the CU size decision of P frame and B frame. Futhermore, the intra CU size fast decision algorithm is to determine the CU size and intra prediction mode beforehand by evaluating the CU texture complexity or on the basis of the CU depth information. The inter CU size fast decision algorithm is to determine the CU size and inter PU mode beforehand on the basis of the neighboring CU depth or the middle encoding parameters. This work, the proposed algorithm mainly focus on the inter CU size fast decision.

Many previous works focus main attention on reducing computation complexity of HEVC encoder. According to different use of information, there are four categories for the CU size fast decision: (1) based on the middle encoding parameters, (2) based on the neighboring CU depth, (3) based on the rate-distortion cost (RD-cost), (4) the synthesis of the former three categories methods. The detailed description of these methods are as follows.

The main idea based on the middle encoding parameters of inter prediction fast selection algorithm is that the middle encoding parameters (such as motion vector (MV), coded block flag (CBF), MV difference(MVD), TU size, SAO and so on) can effectively reflect the CU motion compensation effect of inter prediction. These methods can effectively determine the current CU coding mode by using these encoding parameters. The representative works have: Ahn et al. propose the inter CU fast decision algorithms based on the spatial and temporal encoding informations [46]-[47]. These methods use SAO parameters to estimate the CU texture complexity, and use MV, TU size and CBF to estimate the CU motion complexity. Shen et al. propose a fast CU size decision approach based on bayes rule [48], and the features parameters include sum of absolute transformed difference(SATD) and MV that are used to decide the CU coding mode based on bayes rule. Kim et al. propose an early inter mode termination algorithm [49], this method mainly uses the MVD and CBF informations to check skip mode. Chiang et al. use the zero block detection and MVD informations to early terminate the inter prediction CU split process

[50]. Pan et al. propose a merge mode detection of inter CU size decision [51], and this approach uses the zero block detection and inter motion estimation information to early terminate the merge mode. The above mentioned methods mainly use the middle encoding parameters, and selecting the optimal mode depends on that these parameters reflect the effect of CU coding. However, these parameters can reflect the compensation effect of CU inter prediction. It is a partial solution for sub-CU inter prediction, and it affects the accuracy of CU size fast selection.

The main idea based on the neighboring CU depth of inter CU fast selection algorithm is that it uses the the neighboring CU depth information to deduce the current CU depth range. Because the neighboring CU motion complexity and encoding size are close to the current CU motion complexity and size, it can decide the current CU coding mode fast by using the the neighboring CU encoding informations. The representative works have: Shen et al. propose a fast inter CU depth range selection algorithm[52], and this approach uses the spatial neighboring CU depth informations and temporal neighboring CU depth informations to deduce the current CU depth range. Zhang et al. use the depth similarity of current CTU and neighboring spatial-temporal CTU to zoom out the depth range [53]. Leng et al. propose a fast inter CU depth determination method [54], and this method uses the neighboring spatial-temporal CU depth information in frame level and CU level to skip some CU rate-distortion optimization process. Zhang et al. use the spatial and temporal CU depth information to estimate CTU depth range [55], that can reduce the enconding complexity. Lee et al. propose a fast mode selection algorithm based on neighboring spatial-temporal CU information [56]. Similarly, Correa et al. use spatial-temporal CU depth to estimate the current CU depth [57]. Mu et al. propose a fast CU depth decision scheme to reduce the encoder complexity for HEVC [58]. The method is used to predict the CU depth based on the Support Vector Machine (SVM) model. The robustness above mentioned methods are not high, because it depends on the consistency of the texture complexity and motion characteristics between neighboring CU and the current CU.

The main idea based on the RD-cost of inter CU fast selection algorithm is that the RD-cost can reflect the final motion compensation effect of current CU significantly, and it is able to determine early coding mode by calculating part of the pattern of the RD-cost. The representative works have: Tan et al. propose a fast inter CU selection algorithm [59]. Lee et al. propose a fast inter CU mode selection method based on the RD-cost among encoding process[60], and the RD-cost of skip mode and the optimal RD-cost after executing skip/merge and 2N×2N mode are used to decide the CU encoding mode. Cassa et al. propose an approach to determine whether need to skip or termination of the current encoding process by comparing the RD-cost with the setting threshold[61]. Vanne et al. propose a fast inter mode selection algorithm[62]. This method analyzes the distribution of inter prediction mode and the relationship of different prediction mode, and judgment condition whether PU mode needs to execute is proposed. Zhang et al. propose a RD-cost based fast CU depth decision algorithm [63], and method includes a three-output joint classifier to control the risk of false prediction. The optimal mode above methods depends on the threshold of RD-cost. However, this threshold is relative to the image texture and motion intensity, and the threshold is dynamic variation. Thus, the robustness of these methods are not high. Furthermore, it costs much time to calculate the RD-cost.

Some others synthesis of the former three categories methods are proposed to reduce the encoding complexity. Shen et al. propose a fast inter prediction mode selection algorithm based on the neighboring CU depth information and RD-cost[64]. Correa et al. propose the fast inter CU size decision algorithm based on data mining technology[65]-[66]. These approach use the neighboring CU depth, RD-cost and skip/merge flage to decision CU encoding mode based on decision tree (DT) model. Xiong et al. propose a fast CU size decision for inter predcition[67]-[68], these methods use the relationship of the RD-cost and variances of MV.

As a summary, these methods have advantages and disadvantages. The advantages of the middle encoding parameters method are that the feature extraction is straightforward and it need not the extra computational complexity, while the accuracy of this method is low. The advantages of the neighboring CU depth method are that this method is simple and is easy to achieve, however, it depends on the consistency of the neighboring CU. The advantages of the RD-cost method are that the RD-cost threshold determination method is easy to implement, while the cost of RD-cost calculation is is very time-consuming and the robustness is not high. However, the state-of-the-art algorithms can not achieve a better trade-off between encoding complexity and encoding efficiency. Hence, this work focuses on the trade-off between encoding complexity and encoding efficiency in HEVC encoder, a high efficiency CU (coding unit) size decision algorithm is proposed.

5.2 Observation and Statistical Analysis

Rate distortion optimization (RDO) is important technology for video coding. RDO can achieve a better trade-off between rate and distortion by choosing encoding parameters. In HEVC reference software HM, a two-step RDO method for mode decision. At First, to save computation overhand, a fast RDO is used for early termination, and fast RDO selects the motion vectors and modes for inter prediction. The minimize low-complexity RD-cost function J_{pred} is defined as

$$\min J_{pred} \quad J_{pred} = D_{pred} + \lambda_{pred} \times R_{pred} \tag{5.1}$$

where D_{pred} represents the distortion of between the original block and reference block, R_{pred} represents the number of coding bits, and λ_{pred} is the Lagrange multiplier. Then, a full RDO is used for final decision. The full RDO is determinated by CABAC bit rate and distortion cost, and the minimize full RD-cost function J_{mode}

$$J_{mode} = (SSE_{luma} + \omega_{chroma} \times SSE_{chroma}) + \lambda_{mode} \times R_{mode} \tag{5.2}$$

where SSE_{luma} and SSE_{chroma} respresent the sum of square error (SSE) between the original and reconstructed luma and chroma blocks, R_{mode} represents the number of the coding bits, and ω_{chroma} is weighting factor. However, the cost of full RDO is high in real-time HEVC encoder.

In HEVC, CU is coded using inter prediction or intra prediction, and CU residues are coded using the transform unit (TU). Each TU can be split into four sub-TUs as the residual quad-tree. After the quad-tree structure determines the transform blocks, a syntax element named coded block flag(CBF) is transmitted to indicate whether the TU has non-zero transformed coefficients or not. When a CBF equal one, it indicates that the positions of the non-zero coefficient in the transform block, are coded. In contrary, when a CBF equals zero, all of the transform coefficients in the transform block are zero. Fig.5.1 shows the CBF signaling for a chroma TU, and the root TU CBF bit "1" is signaled.

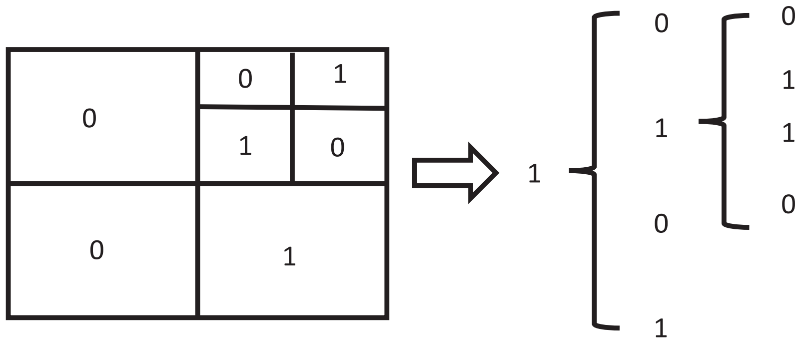

Figure 5.1 CBF for a chroma TU.

The CBF is an important factor for deciding the CU splitting or non-splitting. When CBF equals zero, the image texture tends to be more smooth. Therefore, the current CU tends not to be splitted. On the contrary, when CBF equals one, the image texture tends to be complex. Therefore, the current CU tends to be splitted.

In HEVC, based on RDO (rate-distortion optimization) model, the RD-cost are calculated to decide whether the current CU splits or not. Therefore, the current CU non-splitting or splitting is formulated as a binary classification problem. This two-class event is denoted by discrete random variable y, which is defined as

$$y = \begin{cases} 0,\text{(if CU non-splitting)} \\ 1,\text{(if CU splitting)} \end{cases}$$

Extensive experiments are to investigate the probability distribution of CU splitting or non-splitting with different class sequences, and the results are shown in Table 5.1. The sequences have different resolutions: Traffic(2560×1600), BQTerrace(1920× 1080), Vidyo1(1280×720), BQMall(832×480), and BlowingBubbles(416×240). The profile is LD (lowdelay), and CU size is 64×64 when QP(quantization parameter) is 32. The result is shown that the probability of CU non-splitting is high for high resolution. On the contrary, the probability of CU splitting is high for low resolution. Similar to the same simulation environment, Table 5.2 represents the conditional probability distribution of $p(x_{cbf}|y)$ with different resolution sequences, where x_{cbf}

represents the value of CBF, and $p0 = p(x_{cbf} = 0|y = 0)$, $p1 = p(x_{cbf} = 1|y = 1)$. It can be seen that the conditional probability of $x_{cbf} = 0$ is high for CU non-splitting. On the contrary, the conditional probability of $x_{cbf} = 1$ is high for CU splitting.

Table 5.1 Probability distribution of the CU splitting and Non-splitting.

Sequence Resolution	**Sequence**	**Non-splitting**	**Splitting**
2560 × 1600	**Traffic**	0.60	0.40
1920 × 1080	**BQTerrace**	0.70	0.30
1280 × 720	**Vidyo1**	0.82	0.18
High Res.	**Average**	0.71	0.29
832 × 480	**BQMall**	0.16	0.84
416 × 240	**BlowingBubbles**	0.48	0.52
Low Res.	**Average**	0.32	0.68

Table 5.2 Conditioanal probability distribution of $p(x_{cbf}|y)$.

Sequence Resolution	**Sequence**	$p0$	$p1$
2560 × 1600	**Traffic**	0.96	0.73
1920 × 1080	**BQTerrace**	0.97	0.54
1280 × 720	**Vidyo1**	0.96	0.61
High Res.	**Average**	0.96	0.63
832 × 480	**BQMall**	0.91	0.88
416 × 240	**BlowingBubbles**	0.88	0.70
Low Res.	**Average**	0.90	0.79

In addition, based on some observations from experiments with many sequences, the RD-cost probability density function (*pdf*) of the CU non-splitting and CU splitting are to obey Gaussian distribution. For traffic sequence, Fig.5.2 and Fig.5.3 show the RD-cost conditional probability density function of CU non-splitting and splitting. It can be seen that $p(x_{cost}|y)$ is assumed to follow the Gaussian distribution, where x_{cost} represents the value of RD-cost. Based on the this analysis, we proposed the CU termination and CU skip algorithm using the two features: RD-costs and CBF.

5.3 Proposed CU Pruning Decision for Inter Prediction

In this section, an effective CU size decision algorithm is proposed in the HEVC inter prediction. However, the proposed method is different from Shen's [48] and Lee's [60] work. Firstly, the CBF flag is used to make CU size decision. Secondly, the decision strategy is based on Naive Bayes model. Furthermore, this statistical

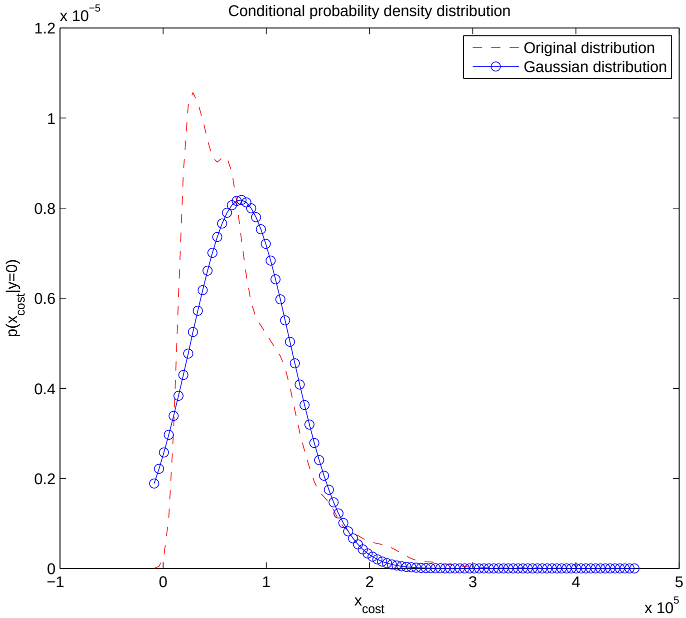

Figure 5.2 The distributions of RD cost of CU non-splitting in Traffic sequence when QP =32.

parameters are estimated by using offline learning and online learning methods.

5.3.1 Fast CU Termination Decision

Thus, for the random variable y of the current CU splitting or non-splitting, the probability density function (*pdf*) $p(y)$ follows a discrete Bernoulli (p) distribution, which is defined as

$$p(y) = \begin{cases} p, \text{ if } y = 0, \\ 1 - p, \text{ if } y = 1. \end{cases}$$

where p is the probability of CU non-splitting.

When we have a classification problem in which the input features x are continuous-valued random variables. Having fit these parameters, to make a prediction on CU non-splitting or splitting with features x, the CU termination decision rule is:

$$\begin{cases} \text{p(y=0|x)} > \text{p(y=1|x), CU non-splitting is made,} \\ \text{else, CU splitting is made .} \end{cases}$$

where the class-conditional probability density function $p(y|x)$ is calculated based Naive Bayes (NB) model

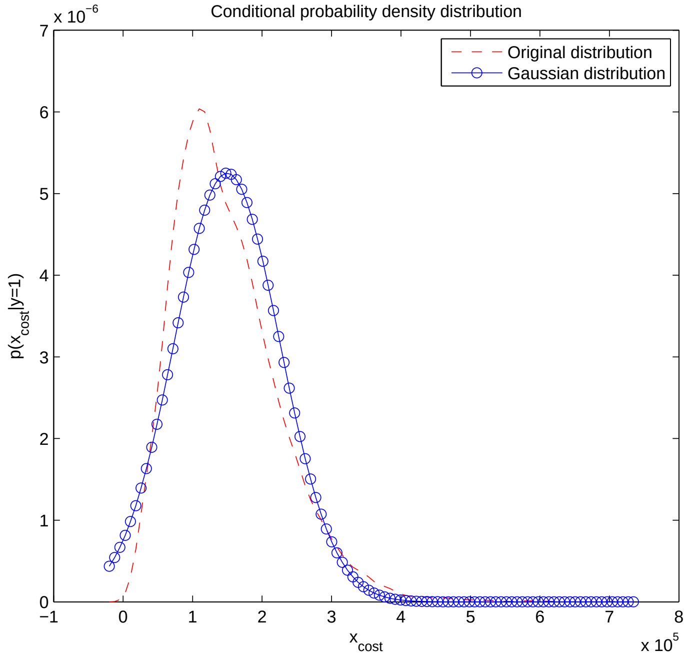

Figure 5.3 The distributions of RD cost of CU splitting in Traffic sequence when QP =32.

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)} = \frac{(\prod_{i=1}^{n} p(x_i|y))p(y)}{(\prod_{i=1}^{n} p(x_i|y=0))p(y=0) + (\prod_{i=1}^{n} p(x_i|y=1))p(y=1)} \tag{5.3}$$

where the features x_i are the coded block flag (CBF) and RD cost of partition $2N \times 2N$, denoted as x_1 and x_2. The prior probability function $p(x_1|y)$ is modeled using a discrete Bernoulli (ϕ) distribution, which are defined as

$$p(x_1|y) = \begin{cases} \phi, \text{ if } y = 0 \ , \\ 1 - \phi \text{ if } y = 1 \ . \end{cases}$$

The prior probability function $p(x_{cost}|y)$ is modeled using the Gaussion distribution. The model is:

$$x_2|y = 0 \sim N(\mu_0, \sigma_0^2)$$

$$x_2|y = 1 \sim N(\mu_1, \sigma_1^2)$$

where the parameters (μ_0, σ_0), (μ_1, σ_1) are mean vectors and covariance matrix of CU non-splitting and splitting,respectively. It is noted that this parameters are estimated by the maximum likelihood estimation. Thus, the prior probability of $p(x_{cost}|y)$ are defined as:

$$p(x_2|y=0) = \frac{1}{\sqrt{2\pi}\sigma_0} e^{-\frac{(x_2-\mu_0)^2}{2\sigma_0^2}} \tag{5.4}$$

$$p(x_2|y=1) = \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(x_2-\mu_1)^2}{2\sigma_1^2}} \tag{5.5}$$

Actually, the evidence probability $p(x)$ is the constant. In order to make a prediction, we update the prior distribution to the posterior:

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)} \propto p(x|y)p(y) = (\prod_{i=1}^{n} p(x_i|y))p(y) \tag{5.6}$$

Thus, the decision function D(y) is defined as:

$$D(y) = (\prod_{i=1}^{n} p(x_i|y))p(y) \tag{5.7}$$

After the introduction of classifier, the CU splitting and non-splitting are decided by Naive Bayes classifier. Therefore, the proposed approach includes two methods: CU early termination (CUET) decision and CU early skip (CUES) decision. CUET decision is to terminate the CU further splitting, and CUES decision is to skip the CU mode selection in the current depth and go to the next depth of the CU.

Thus, when the context of image is tended to smooth, the probability of CU non-splitting is higher than the probability of CU splitting. The condition of CUET decision is that:

$$CUET\ condition \begin{cases} \text{CBF} = 0\ , \\ \text{D(y=0) >D(y=1)} \end{cases}$$

On the other hand, when the context of image is tended to complex, the probability of CU splitting is higher than the probability of CU non-splitting. The condition of CUES decision is that:

$$CUES\ condition \begin{cases} \text{CBF} = 1\ , \\ \text{D(y=1) >D(y=0)} \end{cases}$$

5.3.1.1 Statistical Parameter Estimation

Table 5.3 summarizes the statistical parameters. In our approach, the statistical parameters (mean, standard deviation and prior) are estimated by the offline learning or online learning method.

Table 5.3 The lookup table of estimation parameter.

Learning parameter	**Derived parameter**	**Description**
p	$p(y)$	Probability of CU non-splitting
(μ_i, σ_i)	$p(x_{cost}\|y)$	Conditional probability of RD cost
ϕ	$p(x_{cbf}\|y)$	Conditional probability of CBF

The statistical parameters are estimated by using a non-parametertic estimation with offline learning [67], and are stored in a lookup table (LUT0). It is noticed that the statistical parameters are varied as the CU depth, QP and resolution changed. Thus, we select the sequences: Traffic(2560×1600), BQTerrace(1920×1080), Vidyo4(1280×720), BQMall(832×480), and BlowingBubbles(416×240) for non-parametertic estimation. In the inter prediction stage by using the proposed algorithm, the statistical parameters are indexed by CU depth, QP and resolution.

5.3.1.2 Overall Algorithm

Jointing the CU termination and skip algorithm, the flowchart of presented overall algorithm based on offline learning or online learning is shown as Fig.5.4. This overall algorithm can be divided into five steps.

Step 1: Start CU size decision with motion estimation process.

Step 2: Look up the statistical parameters in LUT0.

Step 3: When the current mode is $2N \times 2N$ and CBF is zero, calculate $D(y = 0)$ and $D(y = 1)$ as formula (7). If $D(y = 0) > D(y = 1)$, it allows terminating all the remaining RD cost computing.

Step 4: When the current mode is $2N \times 2N$ and CBF is 1, If $D(y = 1) > D(y = 0)$, it skips the RD cost computing for remaining CUs in the same depth and goes to the next depth of the CU.

Step 5: If the current depth is less than the maximal depth, depth=depth+1 and repeat step 2, 3. Otherwise, the best CU size is determined.

5.3.2 Improvement CU Size Decision Algorithm

In HEVC, there is correlation between current CU and neighborhood CU. In order to utilize the spatio-temporal correlation, the four neighborhood system M is

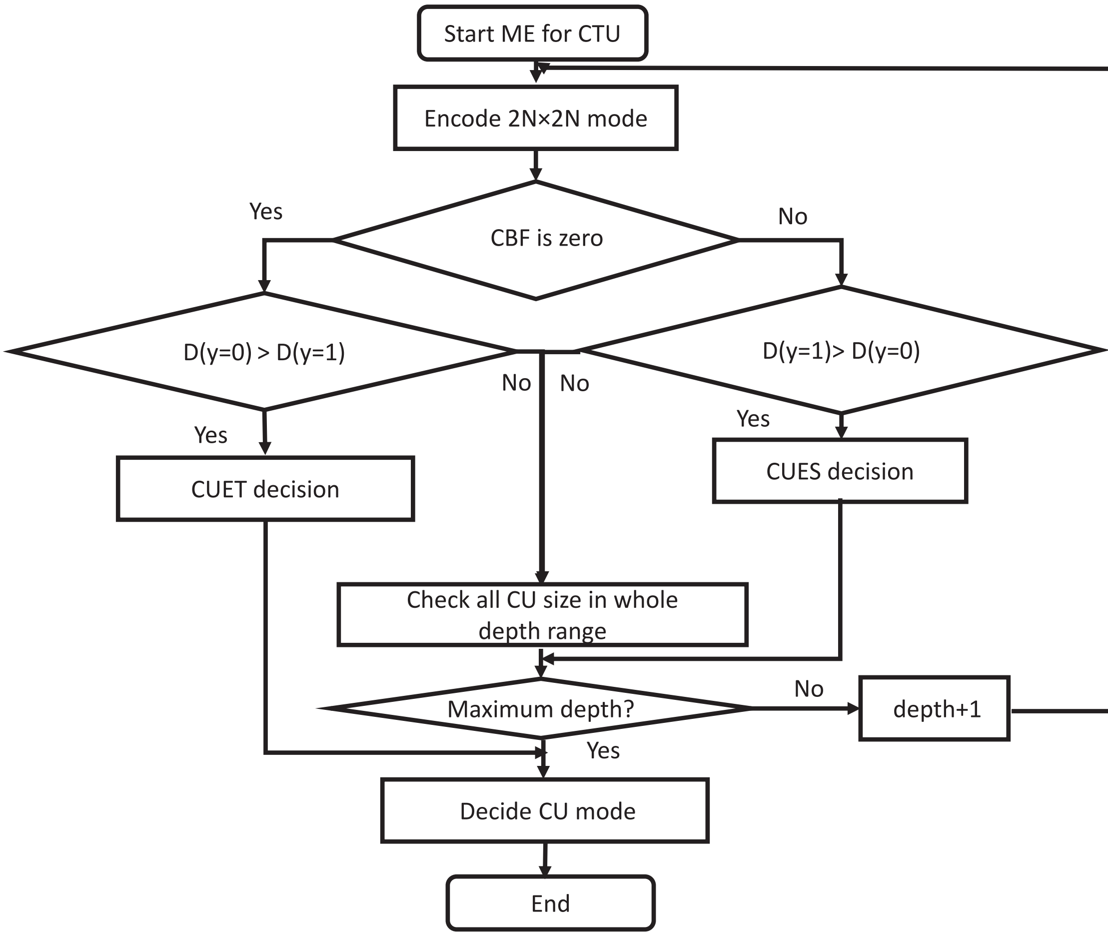

Figure 5.4 Flowchart of the proposed overall algorithm.

defined as

$$M = \{CU_1, CU_2, CU_3, CU_4\}$$

As Fig.5.5 shown, CU_1, CU_2, CU_3 denote the spatially adjacent CUs of the current CU, and CU_4 denote the temporally adjacent CU of the current CU.

Whereas, the prior distribution $p(y)$ can be confirmed by the probabilistic graphical model: Markov Random Fields (MRF)[8] .

$$p(y) = \frac{1}{Z} exp(-\sum_{j \in M} V_j(X_j)) \tag{5.8}$$

From the physicists, this is the Gibbs distribution with interaction potential $\{V_j, j \in M\}$, energy $U = \sum_j V_j$, and partition function of parameters Z. Configurations of lower energies are the more likely, whereas high energies correspond to low probabilities. The CU size decision is a binary classification problem, and the binary

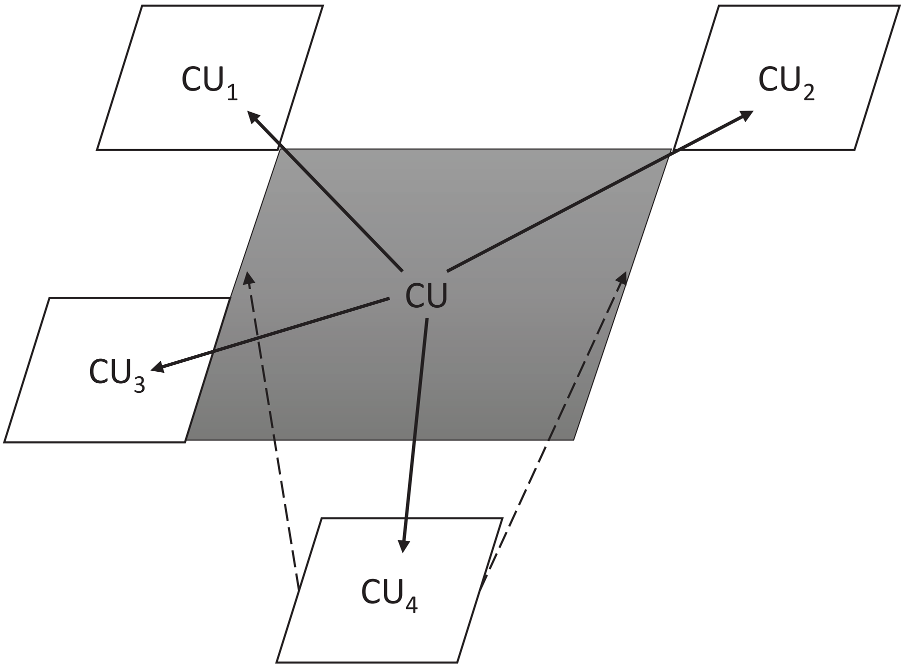

Figure 5.5 The neighborhood system of the current CU.

problem can be modeled by a simple ISING-MRF model[71]:

$$V_j(X_j) = -\beta \times (X_0 \times X_j) \tag{5.9}$$

where the X_0 denotes the flag of the CU_0 splitting or non-splitting, X_j denotes the flag of the CU_j splitting or non-splitting in neighborhood system M, if $X_j = X_0$, $V_j(X_j) = 1$, else $V_j(X_j) = 0$. β is coupling factor, which indicates the strength of CU correlation with neighborhood system M. Thus, the prior $p(y)$ exhibits a factorized form

$$p(y) \propto exp(-\sum_{j \in M} -\beta \times (X_0 \times X_j)) \tag{5.10}$$

Take log funcion of the posterior $p(y|x)$, and it can be written as:

$$\ln p(y|x) \propto [C_1 - \frac{1}{2\sigma_i^2}(x_2 - \mu_i)^2 - \sum_{j \in M} -\beta \times (X_0 \times X_j)] \tag{5.11}$$

where constant $C_1 = \ln p(x_1|y)$, $i = 0, 1$. Then when CU_0 neighborhood system M is valid, the decision function D(y) can be defined as:

$$D(y) = [C_1 - \frac{1}{2\sigma_i^2}(x_2 - \mu_i)^2 - \sum_{j \in M} -\beta \times (X_0 \times X_j)] \tag{5.12}$$

However, when CU_0 neighborhood system M is invalid, the $p(y)$ can be confirmed by Bernoulli(ψ) distribution.The decision function can be rewritten as:

$$D(y) = [C_1 - \frac{1}{2\sigma_i^2}(x_2 - \mu_i)^2 + \ln p(y)] \tag{5.13}$$

Through the above analysis, the improved CU size decision algorithm based on ISING-MRF model includes CUET decision and CUES decision. The condition of CUET decision is: CBF=0, and $D(y = 0) > D(y = 1)$. The condition of CUES decision is: CBF=1, and $D(y = 1) > D(y = 0)$.

5.4 Experiment Results

The proposed algorithm is implemented and verified based on HEVC test model HM12.0. The test conditions are set to evaluate the performance of the proposed algorithm at different profiles (LD and RA). The quantization parameters (QP_i) are set to 22, 27, 32 and 37, respectively. The coupling factor β ($0 < \beta < 1$) indicates the strength of CU correlation with neighborhood system M, and β is set to 0.5 and 0.75 in this work.

The performance of the proposed algorithm is evaluated Bjontegarrd Delta bitrate (BDBR) and peak signal to noise ratio (BDPSNR) [31], and the average time saving (TS) is defined as

$$TS(\%) = \frac{1}{4}\sum_{i=1}^{i=4} \frac{T_{HM}(QP_i) - T_{pro}(QP_i)}{T_{HM}(QP_i)} \times 100\% \tag{5.14}$$

where $T_{HM}(QP_i)$ and $T_{pro}(QP_i)$ are the encoding time by using the HEVC reference software and the proposed method with different QP_i.

5.4.1 Performance of Navie Bayes based CU Size Decision Algorithm

Table 5.4 shows the results of the Naive Bayes based method compared to HEVC reference software. The third and four columns in the table show the performance under the low delay (LD) perfile. From the experimental results can be seen that, in the aspect of encoding complexity, the method can save 50.24% encoding time, while the encoding efficiency can be reduced 1.36%. Thus, in the low delay perfile, this perposed method almost does not affect the encoding quality while the encoding complexity can be reduced significantly.

The five and six columns in the table show the performance under the random access (RA) perfile. In the aspect of encoding complexity, the method can save

Table 5.4 The performance of Naive Bayes method.

		LD		RA	
Class	**Sequence**	**BR(%)**	**TS(%)**	**BR(%)**	**TS(%)**
2560 × 1600	**Traffic**	1.23	55.29	1.63	56.17
	SteamLocomotive	0.40	51.56	0.74	55.86
1920 × 1080	**Kimono**	1.90	41.58	2.35	44.82
	ParkScene	1.15	52.5	1.33	55.77
	Cactus	1.56	47.66	1.78	49.56
	BasketballDrive	4.01	44.66	4.94	47.35
	BQTerrace	0.72	54.08	1.01	54.31
1280 × 720	**Vidyo1**	1.65	64.54	1.70	63.80
	Vidyo3	1.30	61.83	1.41	61.95
	Vidyo4	1.14	65.11	1.46	62.74
High Res.	**Average**	**1.51**	**53.81**	**1.83**	**55.13**
832 × 480	**BasketballDrill**	2.14	45.59	1.97	49.20
	BQMall	1.00	49.20	1.16	56.45
	PartyScene	0.67	39.53	0.96	50.25
	RaceHorses	1.27	36.78	1.79	43.62
416 × 240	**BasketballPass**	1.06	53.02	1.50	54.51
	BQSquare	0.34	47.58	0.65	53.40
	BlowingBubbles	1.50	43.56	1.45	49.32
Low Res.	**Average**	**1.14**	**45.04**	**1.35**	**50.96**
Average		1.36	50.24	1.64	53.48

53.48% encoding time, while the encoding efficiency can be reduced 1.64%. Thus, in the low delay perfile, this perposed method almost does not affect the encoding quality while the encoding complexity can be reduced significantly.

The results demonstrate that the proposed method can significantly reduce the encoding time, and the time saving in high resolution is higher than in the low resolution applications. Furthermore, for 1280 × 720 resolution, the time saving is more than 60% in the LD and RA profiles.

5.4.2 Performance of Improvement CU Size Decision Algorithm

The performance of the improvement CU size decision method is shown as Table 5.5. The third to six columns in the table shows the performance of the improvement method, when β is set to 0.5. The third and four columns in the table show the performance under LD perfile. From the experimental results can be seen that, in the aspect of encoding complexity, the method can save 50.26% encoding time, while the encoding efficiency can be reduced 0.94%. The five and six columns in the table show the performance under the RA perfile. In the aspect of encoding complexity, the method can save 53.48% encoding time, while the encoding efficiency can be reduced 1.64%. Thus, in the low delay perfile, this perposed method almost does not affect the encoding quality while the encoding complexity can be reduced significantly.

The third to six columns in Table 5.6 in the table shows the performance of the improvement method, when β is set to 0.75. The seven and eight columns in the table show the performance under the LD perfile. From the experimental results can be seen that, in the aspect of encoding complexity, the method can save 50.24% encoding time, while the encoding efficiency can be reduced 0.89%. The five and six columns in the table show the performance under the RA perfile. In the aspect of encoding complexity, the method can save 53.48% encoding time, while the encoding efficiency can be reduced 1.14%. Thus, in the low delay perfile, this perposed method almost does not affect the encoding quality while the encoding complexity can be reduced significantly.

Compared with the Table 5.4 and Table 5.6, the encoding complexity of the improvement method is almost the same as the encoding complexity of the Naive Bayes based method, while the encoding efficiency of the improvement is better than the encoding efficiency of the Naive Bayes based method. That is, the improvement method can make a better trade-off between the encoding complexity and encoding efficiency.

To evaluate the steady performance, the R-D curves of the typical sequences are as shown in Fig.5.6 and Fig.5.7 for the LD and RA profiles. It can be noticed that, no matter in high bitrate or in low bitrate, the R-D performance of proposed method is almost similar to the HEVC reference software.

5.4.3 Comparison with Previous Work

Furthermore, the performance of the improvement method is compared with previous works [48],[52], [60],[63], [67] and [68] with $\beta = 0.75$, and the results are shown in Table 5.7. Shen's method [48] is based on the middle encoding parameters of inter prediction. Shen's [52] is based on the neighoring CU depth of inter CU fast selection. Zhang's method [63] and Lee's method [60] are based on the RD-cost of inter CU fast selection. Xiong's methods [67]-[68] are based on the RD-cost and the

Table 5.5 The performance of improvement method.(β=0.5)

		LD		RA	
Class	**Sequence**	**BR(%)**	**TS(%)**	**BR(%)**	**TS(%)**
2560 × 1600	**Traffic**	0.85	54.72	1.07	59.99
	SteamLocomotive	0.22	50.8	0.78	58.39
1920 × 1080	**Kimono**	1.57	41.06	1.83	49.02
	ParkScene	0.99	52.19	1.15	58.54
	Cactus	1.05	46.84	1.36	53.68
	BasketballDrive	2.23	44.32	3.78	51.26
	BQTerrace	0.73	53.76	0.69	57.8
1280 × 720	**Vidyo1**	0.95	64.47	1.87	66.63
	Vidyo3	0.72	61.88	0.63	64.69
	Vidyo4	0.61	64.77	1.01	65.26
High Res.	**Average**	**0.99**	**53.81**	**1.32**	**58.36**
832 × 480	**BasketballDrill**	0.59	46.47	0.72	49.92
	BQMall	0.94	50.31	1.06	55.56
	PartyScene	0.66	41.09	0.78	48.63
	RaceHorses	0.88	37.93	1.40	42.79
416 × 240	**BasketballPass**	1.02	52.83	0.85	53.17
	BQSquare	0.53	47.77	0.56	52.04
	BlowingBubbles	1.47	43.16	1.27	47.90
Low Res.	**Average**	**0.87**	**45.65**	**0.95**	**50.00**
Average		0.94	50.26	1.17	55.02

middle encoding parameters of inter prediction.

It can be seen from the comparison of experimental results that, no matter in high resolution or low resolution, the encoding complexity of the proposed improvement method can be reduced significantly, while the encoding efficiency of the proposed improvement method is better than previous works. Thus, my proposed method can achieve a better trade-off between the encoding complexity and encoding efficiency, compared with state-of-art works.

5.4.4 Hardware Implementation Analysis

In HEVC encoder architecture, the inter prediction engine is the most computation-intensive part. Therefore, unavoidable exhausted iteration of RDO process for each possible CU size becomes the bottle neck. The inherent deep correlations between PUs make it difficult to perform parallel PU inter prediction. Considering the hardware architecture of inter prediction engine, the serial RDO process requires a very high working clock frequency to fulfill real-time encoding.

Fig.5.8 shows a typical architecture of the HEVC encoder with the proposed

Table 5.6 The performance of improvement method.(β=0.75)

		LD		RA	
Class	**Sequence**	**BR(%)**	**TS(%)**	**BR(%)**	**TS(%)**
2560 × 1600	**Traffic**	0.89	55.29	1.06	56.17
	SteamLocomotive	0.22	51.56	0.78	55.86
1920 × 1080	**Kimono**	1.49	41.58	1.84	44.82
	ParkScene	1.07	52.5	1.10	55.77
	Cactus	0.98	47.66	1.36	49.56
	BasketballDrive	2.08	44.66	3.69	47.35
	BQTerrace	0.71	54.08	0.70	54.31
1280 × 720	**Vidyo1**	0.98	64.54	0.86	63.80
	Vidyo3	0.86	61.83	0.59	61.95
	Vidyo4	0.32	65.11	1.09	62.74
High Res.	**Average**	**0.96**	**53.88**	**1.31**	**55.13**
832 × 480	**BasketballDrill**	0.54	45.59	0.74	49.2
	BQMall	0.74	49.20	0.98	56.45
	PartyScene	0.59	39.53	0.75	50.25
	RaceHorses	0.94	36.78	1.39	43.62
416 × 240	**BasketballPass**	0.81	53.02	0.66	54.51
	BQSquare	0.58	47.58	0.55	53.40
	BlowingBubbles	1.33	43.56	1.18	49.32
Low Res.	**Average**	**0.79**	**45.04**	**0.89**	**50.96**
Average		0.89	50.24	1.14	53.48

CU pruning model. The proposed CU pruning model can help the inter prediction engine to select optimal CU size before RDO process. For each $N \times N$ CU, a probability calculator is used to decide CU splitting and non-splitting by using RD-cost and CBF. This probability calculator is based on Naive Bayes model and Gaussion distribution of RD cost. It supports all $N \times N$ CU size for 64×64 to 8×8.

This CU pruning model can reduce redundant CU size directly with very low hardware cost. Different from a hard threshold, this proposed method can achieve the tradeoff between encoding efficiency and encoding complexity. By reducing the redundant RDO iteration the working clock frequency can be decreased and low cost and low power hardware architecture can be easily achieved. Furthermore, the proposed CU pruning model can work well with other efficient inter prediction algorithm easily.

5.5 Summary of this chapter

This work focuses on balancing the trade-off between encoding efficiency and encoding complexity in HEVC inter prediction. A high efficiency CU (coding unit) size

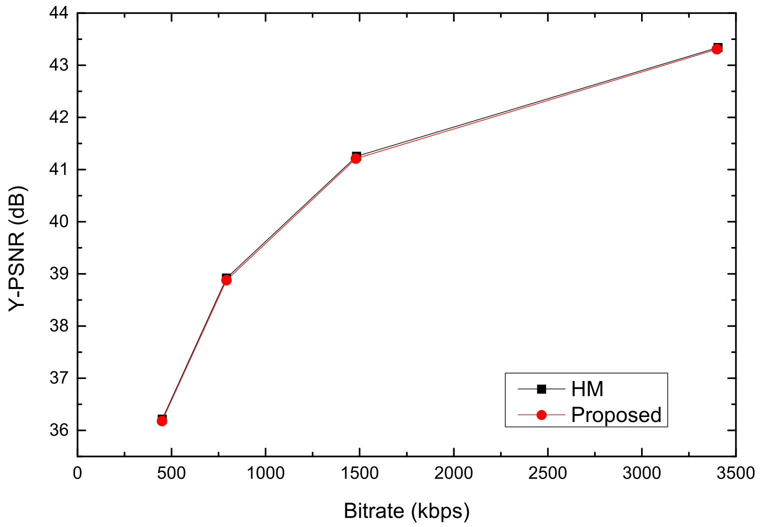

Figure 5.6 R-D curve of the improvement method (β=0.75, Vidyo4, Low Delay)

Table 5.7 Performance of the proposed method compared with previous work.

Config	Method	(BR, TS) High Res.	Low Res.	Average
LD	**Pro.(β= 0.75)**	**(0.96, 53.88)**	**(0.79, 45.04)**	**(0.89, 50.24)**
	Shen's[52]	(0.97, 47.11)	(1.33, 33.89)	(1.15, 41)
	Lee's [60]	(1.31, 69)	(1.13, 53)	(1.22, 61)
	Zhang's[63]	(2.41, 62.59)	(1.55, 40.31)	(1.98, 51.45)
	Xiong's[67]	(2.78, 66.13)	(1.6, 53.21)	(2.19, 59.67)
	Xiong's[68]	(2.59, 44.40)	(1.83, 36.26)	(2.21, 40.33)
RA	**Pro.(β= 0.75)**	**(1.31, 55.13)**	**(0.89, 50.96)**	**(1.14, 53.48)**
	Shen's[48]	(1.25, 51.05)	(1.33, 38.61)	(1.35, 44.7)
	Shen's[52]	(1.30, 45.25)	(1.65, 38.78)	(1.49, 42)
	Lee's[60]	(1.49, 65.43)	(1.37, 58.57)	(1.43, 62)
	Xiong's[67]	(3.3, 69.24)	(2.18, 57.10)	(2.74, 63.17)

decision algorithm is proposed for HEVC inter coding, which contains two methods: CU size early termination (CUET) decision method and CU size early skip (CUES) decision method. The CU pruning is modeled as a binary classification problem based on Naive Bayes (NB) model. Furthermore, an improvement method based Markov Random Fields (MRF) model is proposed to improve the algorithm performance. The difference from previous works is that residual flag in inter-coded CU is used to predict as a feature. The offline learning method is used to obtain the statistical parameters. The proposed algorithm can significantly reduce the encoding complexity. Furthermore, it can bring the low power cost in the hardware imple-

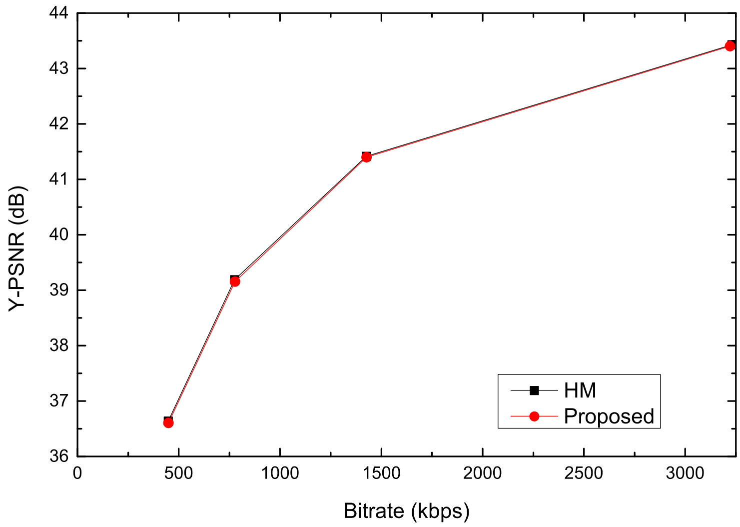

Figure 5.7 R-D curve of the improvement method (β=0.75, Vidyo4, Random Access)

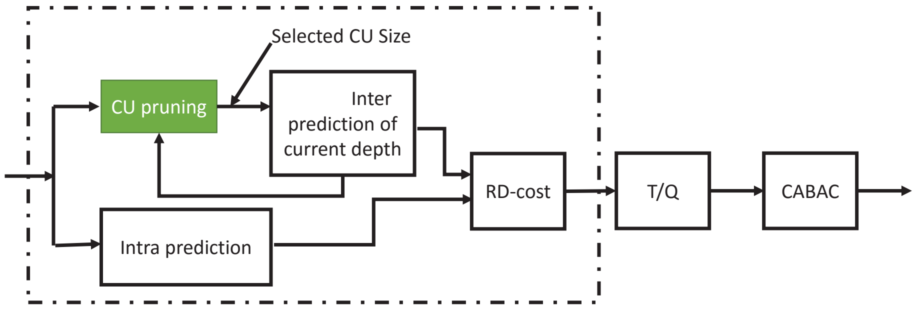

Figure 5.8 Architecture design of HEVC encoder.

mentation. The simulation experiment results demonstrate that the improvement method can significantly reduce 53.48% and 50.24% average encoding complexity under random access and low delay profiles, while the encoding efficiency can be reduced 1.14% and 0.89% on average. Moreover, the RD performance of proposed method is similar to HEVC reference software.

Chapter 6 Conclusion

In this dissertation, the basic technology spcified in the H.265/HEVC standard are investigated. In order to face the future video coding challenge, three methods are proposed to process the encoding in parallel, improve the high efficiency and reduce the encoding complexity in encoder.

In chapter 2, the basic encoding tools of coding structures, intra prediction, inter prediction and transform and quantization, in-loop filters and entropy encoding are investigated. Moreover, the main challenge of the H.265/HEVC is investigated. There are mainly three requirements for video compression: parallel processing, high compression capability, and low computational complexity.

In chapter 3, a spatio-temporal prediction algorithm is proposed to improve the parallelism of motion estimation in HEVC. Firstly, using the character of the encoding block, a spatial prediction algorithm is presented. Secondly, a temporal prediction algorithm is proposed to remove the dependency of encoding block, which make to determine the temporal motion vector only once within a PMER. At last, a joint spatial and temporal prediction algorithm is proposed to improve the encoding complexity in parallel.

In chapter 4, firstly, an efficient motion vector prediction selection algorithm is presented. The MVs surrounding the coding unit are used to get the accurate PMV. After that, the search range is adjusted adaptively in motion estimation process.

In chapter 5, this work focuses on balancing the trade-off between encoding efficiency and encoding complexity in HEVC inter prediction. A high efficiency CU (coding unit) size decision algorithm is proposed for HEVC inter coding, which contains two methods: CU size early termination (CUET) decision method and CU size early skip (CUES) decision method. The CU pruning is modeled as a binary classification problem based on Naive Bayes (NB) model. Furthermore, an improvement method based Markov Random Fields (MRF) model is proposed to improve the algorithm performance. The difference from previous works is that residual flag in inter-coded CU is used to predict as a feature. The offline learning method is used to obtain the statistical parameters. The proposed algorithm can significantly reduce the encoding complexity.

The proposals mentioned in this dissertation can be further used in the whole H.265/HEVC encoder to improve the encoding performance especially for the real-time applications of high-definition video, digital cinema and multiview video.

In the future, the hardware architecture based on these approaches is designed. Furthermore, the parallel processing and low complexity approachs based future video coding technology will be explored.

Bibliography

[1] G.J. Sullivan, J.R. Ohm, W.J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp. 1649-1668, Dec. 2012.

[2] 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. 2003.

[3] J.M. Boyce, Y. Ye, J. Chen, and A.K. Ramasubramonian, "Overview of SHVC: scalable extensions of the high efficiency video coding (HEVC) standard,". IEEE Trans. Circuits Syst. Video Technol., vol. 26, no. 1, pp. 20-34. 2016.

[4] G. Tech, Y. Chen, K. Muller, J. R. Ohm, A. Vetro, and Y. K. Wang, "Overview of the multiview and 3D extensions of high efficiency video coding,". IEEE Trans. Circuits Syst. Video Technol., vol. 26, no. 1, pp. 35-49. 2016.

[5] J.Z. Xu, R. Joshi, R.A. Cohen, "Overview of the emerging HEVC screen content coding extension,". IEEE Trans. Circuits Syst. Video Technol., vol. 26, no. 1, pp. 50-62. 2016.

[6] K. Inoue, K. Isechi, H. Saito and Y. Kuroki, "An inter-prediction method using sparse representation for high efficiency video coding," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol. E96-A, no. 11, pp. 2191-2193, Nov. 2013.

[7] G. Tian, X. Jin and S. Goto, "Content adaptive hierarchical decision of variable coding block sizes in high efficiency video coding for high resolution videos," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol. E96-A, no. 4, pp. 780-789, Apr. 2013.

[8] C.C. Chi, M. Alvarez-Mesa, B. Juurlink, G. Clare, F. Henry, S. Pateux, and T. Schierl, "Parallel scalability and efficiency of HEVC parallelization approaches," IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp. 1827-1838, Dec. 2012.

[9] G. Correa, P. Assuncao, L. Agostini, L.A. da Silva Cruz, "Performance and computational complexity assessment of high-efficiency video encoders," IEEE

Trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp. 1899-1909, 2012.

[10] F. Bossen, B. Bross, K. Suhring, and D. Flynn, "HEVC complexity and implementation analysis," IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp. 1685-1696, 2012.

[11] T. Schierl, V. George, A. Henkel, D. Marpe, "Dependent slices," Joint Collaborative Team on Video Coding (JCT-VC), Document JCTVC-I0229, Geneva, Apr.-May 2012.

[12] K. Misra, A. Segall, M. Horowitz, S. Xu, A. Fuldseth, and M. Zhou, "An overview of tiles in HEVC," IEEE Journal of Selected Topics in Signal Processing, vol.7, no.6, pp. 969-977, 2013.

[13] ITU-T Rec. H.265 and ISO/IEC 23008-10. High efficiency video coding. 2013.

[14] J. Lainema, F. Bossen, W.J. Han, J. Min and K. Ugur, "Intra coding of the HEVC standard," IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp. 1792-1801, 2012.

[15] V. Sze, M. Budagavi, G.J. Sullivan. High efficiency video coding (HEVC)(algorithms and architectures). Springer, 2014: 1-375.

[16] M. Wien. High efficiency video coding(Signals and Communication Technology). Springer, 2015: 1-314.

[17] J. Sole, R. Joshi, N. Nguyen, T.Y. Ji, M. Karczewicz, G. Clare, F. Henry, and A. Duenas, "Transform coefficient coding in HEVC," IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp. 1765-1777, 2012.

[18] A. Norkin, G. Bjontegaard, A. Fuldseth, M. Narroschke, M. Ikeda, K. Andersson, and G.V.D. Auwera, "HEVC deblocking filter," IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp. 1746-1754, 2012.

[19] V. Sze, M. Budagavi, "High throughput CABAC entropy coding in HEVC," IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp. 1778-1791, 2012.

[20] X.T. Jiang, T. Song, T. Shimamoto and L.S. Wang, "high efficiency video coding HEVC motion estimation parallel algorithms on GPU," 2014 IEEE International Conference on Consumer Electronics - Taiwan (IEEE ICCE-Taiwan 2014), pp. 115-116, May, 2014.

[21] M.E. Sinangil, V. Sze, M.H. Zhou and A.P. Chandrakasan, "Cost and coding efficient motion estimation design considerations for high efficiency video coding (HEVC) standard," IEEE Journal on Selected Topics in Signal Processing, vol. 7, no. 6, pp. 1017-1028, Dec. 2013.

[22] M.E. Sinangil, A.P. Chandrakasan, V. Sze and M.H. Zhou, "Hardware-aware motion estimation search algorithm development for high-rfficiency video coding (HEVC) standard," 2012 IEEE International Conference on Image Processing (ICIP 2012), pp. 1529-1532, Sept. 2012.

[23] G. Sanchez, B. Zatt, M. Porto and M. Porto, "Hardware-friendly HEVC motion estimation: new algorithms and efficient VLSI designs targeting high definition videos," Analog Integrated Circuits and Signal Processing, pp. 1-12, June 2014.

[24] J.L. Lin, Y.W. Chen, Y.W. Huang, and S.M. Lei, "Motion vector coding in the HEVC standard," IEEE Journal of Selected Topics in Signal Processing," vol. 7, no. 6, pp. 957-968, Dec. 2013.

[25] J. Park, S. Park and B. Jeon, "Improvement on median motion vector of AMVP," Document JCTVC-D095, Jan. 2011.

[26] K. Kazui, S. Shimada, J. Koyama, and A. Nakagawa, "Improvement on simplified motion vector prediction," Document JCTVC-E062, Mar. 2011.

[27] J. Zhang, F. Dai, Y.K. Ma and Y.D. Zhang, "Highly parallel mode decision method for HEVC," 2013 Picture Coding Symposium, San Jose, CA, pp. 281-284, 2013.

[28] Q. Yu, L. Zhao, and S.W. Ma, "Parallel amvp candidate list construction for HEVC," 2012 IEEE Visual Communications and Image Processing (VCIP), pp. 1-6, Nov. 2012.

[29] C.G. Yan, Y.D. Zhang, J.Z. Xu, F. Dai, J. Zhang, Q.H. Dai, and F. Wu, "Efficient Parallel Framework for HEVC Motion estimation on many-core processors," IEEE Trans. Circuits Syst. Video Technol., vol. 24, no. 12, pp. 2077-2089, Dec. 2014.

[30] L. Q. Shen, Z. Liu, X. P. Zhang, W. Q. Zhao, and Z. Y. Zhang, "An effective CU size decision method for HEVC encoders," IEEE Transactions on Multimedia,

vol. 15, no. 2, pp. 465-470, Feb. 2013.

[31] G. Bjontegaard, “Calculation of average PSNR differences between RD-curves, ITU-T SG16 Q.6, VCEG-M33, Apr. 2001.

[32] G. Laroche, J. Jung, B. Pesquet-Popescu, “RD optimized coding for motion vector predictor selection ”, IEEE Transactions on Circuits and Systems for Video Technology, vol.18, no.9, pp.1247-1257, 2008.

[33] H. Li, B. Li, J. Xu, “Rate-distortion optimized reference picture management for high efficiency video coding ”, IEEE Transactions on Circuits and Systems for Video Technology, vol.22, no.12, pp.1844-1857, 2012.

[34] W. H. Peng, C. C. Chen, “An interframe prediction technique combining template matching prediction and block-motion compensation for high-efficiency video coding”, IEEE Transactions on Circuits and Systems for Video Technology, vol.23, no.8, pp.1432-1446, 2013.

[35] I. K. Kim, S. Lee, M. S. Cheon, T. Lee, J. Park, “Coding efficiency improvement of HEVC using asymmetric motion partitioning, 2012 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)”, pp.1-4, June, 2012.

[36] W. D. Chien, K. Y. Liao, J. F. Yang, “Enhanced AMVP mechanism based adaptive motion search range decision algorithm for fast HEVC coding”, 2014 IEEE International Conference on Image Processing (ICIP), pp.3696-3699, October, 2014.

[37] C.C. Lou, S.W. Lee, C.C. Jay Kuo,“Adaptive motion search range prediction for video encoding”, IEEE Transactions on Circuits and Systems for Video Technology, vol.20, no.12, pp.1903-1908, 2010.

[38] Z. Shi, W. A. C. Fernando, A. Fernando,“Adaptive direction search algorithms based on motion correlation for block motion estimation”, IEEE Transactions on Consumer Electronics, vol.57, no.3, pp.1354-1361, 2011.

[39] Z. P. Deng, Y. L. Chan, K. B. Jia, C. H. Fu, W. C. Siu, “Fast motion and disparity estimation with adaptive search range adjustment in stereoscopic video coding”, IEEE Transactions on Broadcasting, vol.58, no.1, pp.24-33, 2012.

[40] Y. H. Ko, H. S. Kang, S. W. Lee, "Adaptive search range motion estimation using neighboring motion vector differences", IEEE Transactions on Consumer Electronics, vol.57, no.2, pp.726-730, 2011.

[41] A. Paul, "Adaptive search window for high efficiency video coding", Journal of Signal Processing Systems, vol.79, no.3, pp.257-262, 2013.

[42] T. K. Lee, Y. L. Chan, W. C. Siu,"Depth-based adaptive search range algorithm for motion estimation in HEVC", 2014 19th International Conference on Digital Signal Processing (DSP), pp. 919-923, August, 2014.

[43] X.T. Jiang, T. Song, T. Shimamoto, W. Shi and L.S. Wang, "Spatio-Temporal Prediction Based Algorithm for Parallel Improvement of HEVC ", IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol.E98-A, no.11,pp. 2229 -2237,Nov. 2015.

[44] F. Bossen, "Common test conditions and software reference configurations", JCTVC-L1100, Jan. 2013.

[45] G.J. Sullivan, J.R. Ohm, J.R. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard", *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649-1668, 2012.

[46] Ahn S, Kim M, Park S. Fast decision of CU partitioning based on SAO parameter, motion and PU/TU split information for HEVC[C]//Picture Coding Symposium (PCS), 2013. IEEE, 2013: 113-116.

[47] S. Ahn, B. Lee, and M. Kim, "A novel fast CU encoding scheme based on spatiotemporal encoding parameters for HEVC inter coding", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 3, pp. 422-435, Mar. 2015.

[48] X. Shen, L. Yu, and J. Chen, "Fast coding size selection for HEVC based on bayesian decision rule", *in proc. picture coding symposium(PCS)*, pp. 453-456, May, 2012.

[49] Kim J, Yang J, Won K, et al. Early determination of mode decision for HEVC[C]//Picture Coding Symposium (PCS), 2012. IEEE, 2012: 449-452.

[50] Chiang P T, Chang T S. Fast zero block detection and early CU termination

for HEVC video coding[C]//Circuits and Systems (ISCAS), 2013 IEEE International Symposium on. IEEE, 2013: 1640-1643.

[51] Pan Z, Kwong S, Sun M T, et al. Early MERGE mode decision based on motion estimation and hierarchical depth correlation for HEVC[J]. Broadcasting, IEEE Transactions on, 2014, 60(2): 405-412.

[52] L. Shen, Z. Liu, X. Zhang, W. Zhao, and Z. Zhang, "An effective CU size decision method for HEVC encoders", *IEEE Transactions on Multimedia*, vol. 15, pp. 465-470, Feb. 2013.

[53] Zhang Y, Wang H, Li Z. Fast coding unit depth decision algorithm for interframe coding in HEVC[C]//Data Compression Conference (DCC), 2013. IEEE, 2013: 53-62.

[54] Leng J, Sun L, Ikenaga T, et al. Content based hierarchical fast coding unit decision algorithm for HEVC[C]//Multimedia and Signal Processing (CMSP), 2011 International Conference on. IEEE, 2011, 1: 56-59.

[55] Zhang Y, Wang H, Li Z. Fast coding unit depth decision algorithm for interframe coding in HEVC[C]//Data Compression Conference (DCC), 2013. IEEE, 2013: 53-62.

[56] Lee J H, Park C S, Kim B G, et al. Novel fast PU decision algorithm for the HEVC video standard[C]//Image Processing (ICIP), 2013 20th IEEE International Conference on. IEEE, 2013: 1982-1985.

[57] Correa G, Assuncao P, Agostini L, et al. Coding tree depth estimation for complexity reduction of HEVC[C]//Data Compression Conference (DCC), 2013. IEEE, 2013: 43-52.

[58] F. Mu, L. Song, X. Yang, and Z. Luo, "Fast coding unit depth decision for HEVC", *IEEE Int. Conf. Multimedia and Expo Workshops*, pp. 1-6, Jul. 2014.

[59] Tan H L, Liu F, Tan Y H, et al. On fast coding tree block and mode decision for high-efficiency video coding (HEVC)[C]//Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on. IEEE, 2012: 825-828.

[60] J. Lee, S. Kim, K. Lim and S. Lee, "A fast CU size decision algorithm for

HEVC", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 3, pp. 411-421, Mar. 2015.

[61] Cassa M B, Naccari M, Pereira F. Fast rate distortion optimization for the emerging HEVC standard[C]//Picture Coding Symposium (PCS), 2012. IEEE, 2012: 493-496.

[62] J. Vanne, M. Viitanen, and D. Hamalainen, "Efficient mode decision schemes for HEVC inter prediction", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no.9, pp. 1579-1593, Sept. 2014.

[63] Y. Zhang, S. Kwong, X. Wang and H. Yuan, "Machine learning based coding unit depth decisions for flexible complexity allocation in high efficiency video coding", *IEEE Transactions on image processing*, vol. 24, no. 7, pp. 2225 - 2238, July, 2015.

[64] L. Shen, Z. Zhang, and Z. Liu, "Adaptive inter-mode decision for HEVC jointly utilizing inter-level and spatio-temporal correlations", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 10, pp. 1709-1722, Oct. 2014.

[65] G. Correa, P. Assuncao, L. Agostini and L. da, "Fast HEVC encoding decisions using data mining", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 4, pp. 660-673, Apr. 2015.

[66] Correa G, Assuncao P, Agostini L, et al. A method for early-splitting of HEVC inter blocks based on decision trees[C]//Signal Processing Conference (EUSIPCO), 2014 Proceedings of the 22nd European. IEEE, 2014: 276-280.

[67] J. Xiong, H. Li, F. Meng, S. Zhu, Q. Wu, and B. Zeng, "MRF-based on fast HEVC inter CU decision with the variance of absolution differences", *IEEE Transactions on Multimedia*, vol. 16 no. 8, pp. 2141-2153, Dec. 2014.

[68] J. Xiong, H. Li, Q.B. Wu and F. Meng, "A fast HEVC inter CU selection method based on pyramid motion divergence", *IEEE Transactions on Multimedia*, vol. 16 no. 2, pp. 559-564, Feb. 2014.

[69] D. Chai, S.L. Phung, A. Bouzerdoum, "A bayesian skin/non-skin color classifier using non-parametric density estimation", *Proceedings of the 2003 International*

Symposium on Circuits and Systems (ISCAS), vol.2, pp. II-464-II-467, 2003.

[70] J. Yang, J. Kim, K. Won, H. Lee, and B. Jeon “Early skip detection for HEVC”, JCTVC-G543, Geneva, CH, Nov. 2011.

[71] P. Patrick, “Markov random fileds and images”, *IRISA*, 1998.

Acknowledgement

Time goes fast, and I am finishing the three years' studies and lives in Tokushima University for pursuing my PH.D degree. Finally, it is the time to graduate and give acknowledgements to those who ever helped me in my study and life.

First of all, I would like to express my gratitude to my advisors, Associate Professor Song Tian and Professor Shimamoto Takashi, for giving me continuous inspiration, support and criticism throughout the whole of my work. They always provided me with valuable insight and making sure that I was not lost in the research directions. Without their supports, it would be impossible for me to finish this work.

I have had the pleasure of study in the course of electrical and electronic systems, Tukushima University. I would like to extend my appreciation to the Professor Hashizume Masaki, the Professor Nishio Yoshifumi, the Associate Professor Yotsuyanagi Hiroyuki, the Assistant Professor Uwate Yoko, and other technical staffs in our course, for their supports and friendship. I have had the pleasure of collaborating with numerous exceptionally talented graduate students and colleagues over the last few years. I would like to thank all my colleagues in our lab.

Finally, I would like to express my gratitude towards my parents for their unconditional supports and understanding over the years. My deepest love and gratitude go to my wife, Liyuan Zhang for her love, patience, and enduring support. My deepest love go to my cute son, not to accompany you grow together, I am deeply sorry.

Department of Electrical and Electronic Engineering, College of Systems Innovation Engineering, Graduate School of Advanced Technology and Science, The University of Tokushima, Japan

Xiantao Jiang

Appendix

A Publication Paper

A.1 Xiantao Jiang, Tian Song, Takashi Shimamoto, Wen Shi and Lisheng Wang: " Spatio-Temporal Prediction Based Algorithm for Parallel Improvement of HEVC " , IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Vol.E98-A, No.11,pp. 2229 -2237,Nov. 2015.

A.2 Xiantao Jiang, Tian Song, Wen Shi,Takashi Shimamoto and Lisheng Wang: " High Efficiency CU Depth Prediction Algorithm for High Resolution Applications of HEVC " , IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Vol.E98-A,No.12,pp.2528-2536,Dec. 2015.

A.3 Xiantao Jiang, Tian Song, Wen Shi, Takafumi Katayama, Takashi Shimamoto and Lisheng Wang: " Fast CU size decision based on probabilistic graphical model in HEVC inter prediction " , IEICE Transaction on Information and Systems, 2016, 99(11).(Accepted).

A.4 Wen Shi, Xiantao Jiang, Tian Song and Takashi Shimamoto: " Edge Detector Based Fast Level Decision Algorithm for Intra Prediction of HEVC " , Journal of Signal Processing, Vol.19, No.2, pp.67–73, 2015.

B International Conference

B.1 Xiantao Jiang, Tian Song, Takashi Shimamoto, Wen Shi and Lisheng Wang:"Merge Mode Prediction Algorithm for Adaptive Parallel Improvement of HEVC " , Proceedings of IEEE International Conference on Consumer Electronics - Taiwan (2015 ICCE-Taiwan), pp. 310-311, Taipei, June, 2015.

B.2 Xiantao Jiang, Tian Song, Jenq-Shiou Leu,and Takashi Shimamoto: "High Efficiency Inter CU Size Prediction Algorithm for HEVC", International Forum on Advanced Technologies(IFAT2015), Tokushima, Japan, pp. 165-167, March, 2015.

B.3 Xiantao Jiang, Tian Song, Takashi Shimamoto and Lisheng Wang:"AMVP Prediction Algorithm for Adaptive Parallel Improvement of HEVC",The 12th of the biennial Asia Pacific Conference on Circuits and Systems(IEEE APCCAS 2014), Okinawa, Japan, pp.511—514, Nov, 2014.

B.4 Xiantao Jiang, Tian Song, Takashi Shimamoto, Wen Shi and Lisheng Wang:"Temporal Prediction Improvement for Parallel Processing of HEVC",The 12th of the biennial Asia Pacific Conference on Circuits and Systems(IEEE APCCAS 2014), Okinawa, Japan, pp.515–518, Nov, 2014.

B.5 Xiantao Jiang, Tian Song, Takashi Shimamoto and Lisheng Wang: " High Efficiency Video Coding (HEVC) Motion Estimation Parallel Algorithms on GPU",Proceedings of IEEE International Conference on Consumer Electronics - Taiwan (2014 ICCE-Taiwan), pp.115–116, Taipei, May 2014.

B.6 Xiantao Jiang, Tian Song, Wen Shi, Takashi Shimamoto and Lisheng Wang: "High Efficiency CU Depth Decision Algorithm for High Resolution Application of HEVC " , IEEE-TENCON 2015, pp. 1-4, Macau, Nov, 2015.

B.7 Wen Shi, Xiantao Jiang, Tian Song, Jenq-Shiou Leu and Takashi Shimamoto : " Spatial Locality Based Parallel Scheme for Intra Coding of HEVC", Tenth International Conference on Innovative Computing, Information and Control (ICICIC2015), p.204, Dalian, Aug. 2015.

B.8 Wen Shi,Xiantao Jiang, Tian Song and Takashi Shimamoto: "Segmental Down-

sampling Intra Coding Based on Spatial Locality for HEVC", 2015 IEEE Fifth International Conference on Consumer Electronics - Berlin (ICCE-Berlin), pp.12—16, Berlin, sept.2015.

B.9 Jin Ikegita,Xiantao Jiang, Tian Song, Takashi Shimamoto, and Jenq-Shiou Leu: " Efficient Prediction Motion Vector Candidate Selection Algorithm for HEVC " , The 30th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC2015), pp.402–403, Seoul, Jun. 2015.

B.10 Wen Shi, Xiantao Jiang, Tian Song and Takashi Shimamoto: "Efficient Intra Coding for HEVC Based on Spatial Locality", International Forum on Advanced Technologies(IFAT2015), Tokushima, Japan, pp. 168-170, March, 2015.

B.11 Wen Shi, Xiantao Jiang, Tian Song and Takashi Shimamoto: " Spatial Locality Based Supplemental Modes for Intra Prediction of HEVC ",Proceedings of IEEE International Conference on Consumer Electronics - Taiwan (2015 ICCE-Taiwan), pp. 298-299, Taipei, June, 2015.

B.12 Wen Shi,Xiantao Jiang, Tian Song and Takashi Shimamoto: "Edge Information Based Fast Selection Algorithm for Intra Prediction of HEVC ", Proceedings of IEEE Asia Pacific Conference on Circuits and Systems (APCCAS2014), pp.17–20, Okinawa, Nov. 2014.

B.13 Wen Shi,Xiantao Jiang, Tian Song and Takashi Shimamoto: "Edge Detector Based Fast Level Decision Algorithm for Intra Prediction of HEVC " , Proceedings of International Workshop on Nonlinear Circuits, Communications and Signal Processing (NCSP'14), pp.129–132, Honolulu, Mar. 2014.

B.14 Wen Shi, Xiantao Jiang, Tian Song, Jenq-Shiou Leu and Takashi Shimamoto : "Downsampled Information Based Low Complexity Intra Coding for HEVC", Proceedings of 2nd International Forum on Advanced Technologies (IFAT2016), No.P1-17, pp.1–3, Tokushima, Mar. 2016.