Key Extension Technologies for Future Video Coding

Doctor’s Course in Electrical and Electronic Engineering

Graduate School of Engineering, Tokushima University

片山　貴文

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 multi-view 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 corresponding algorithms have increased the difficulty of implementation. Especially, for high-resolution video, multi-view video applications, simulcast etc, software coding require a very high computational complexity. Therefore, high speed, application specified hardware has been acknowledged as a good way of implementation for the H.265/HEVC encoding. Moreover, because the content is increased to satisfy the requirement of the consumer, the importance of the extension technique become very high.

This paper studies on the key technology for the future video coding. As shown in following contents, the main research contents include three parts:

(1) Scalable extension

Recently, various video streams needed to be generated because of increasing of multi-cast service and terminal devices. One approach to full this requirement is coding the video streams in all available formats (simulcast coding) and transmitting them separately. As well know, this approach requires more bandwidth. The other approach has been developed as a scalable video coding (SVC) tool, which enables the video coding system to deliver different versions of the same video content within the same bit-stream. Compared to simulcast coding, SVC requires less bandwidth. Furthermore, many consumer electronic devices are developed for a various display, processing, and transmission capabilities. Therefore, scalable coding plays the role of an important tool.

(2) 3D extension

With the development of the technology of 3D television (3DTV) and free viewpoint television (FTV), 3D video coding (3D-HEVC) attracts more attention. The typical 3D video is represented using the multi-view video and depth format, in which few captured texture videos as well as associated depth maps are used. The depth maps provide per-pixel with the depth corresponding to the texture video that can be used to render arbitrary virtual views by using depth image based rendering. Recently, 3D-HEVC technology based on high efficiency video coding (HEVC) is now being standardized by joint collaborative team on 3D video coding (JCT-3V) as an extension to HEVC. From the JCT-3V meetings, the developed coding schemes for 3D-HEVC mainly use HEVC together with exploiting temporal and interview correlation. Thus, many coding tools applied in 3D-HEVC are based on the hybrid coding scheme and highly related to HEVC.

(3) Future extension

Various extension technologies of HEVC were examined from 2014 to 2016 in order to satisfy consumers' demands. Also, since the precision of image recognition by a artificial intelligence has greatly improved in 2016, to apply the artificial intelligence (AI) to image processing was expected. In other words, it is necessary to propose a new coding technique using AI for super-resolution such as 8K. One of our research aim is to develop the preprocessing software architecture of AI for greatly improving the coding efficiency.

Our research target is the complexity reduction of the extension model of HEVC. We consider that the diverse terminal device, 3D view (VR), and the device including the AI require the complexity reduction and the improvement of the coding efficiency according to extension technology because the target of the future video coding is super-resolution (8K). In other words, the importance of the extension technology is increased for the next generation video coding. The main research contents consist of two conventional extension models and next future model. The main research results are as following:

(1) Scalable extension

Our research focus is on developing a complexity reduction scheme for spatial scalable SHVC encoder. The proposed algorithm uses fast CU depth decision (FCDD), fast mode decision (FMD), and early termination process (ETP). The performance of the proposed algorithm was tested over a representative set of video sequences and was compared to the unmodified SHVC encoder as well as two of the art complexity reduction schemes and combinations. Performance evaluations show that our proposed algorithms reduce encoding time on average 61.88% and increases BD-rate about 0.9%, compared with SHM 11.0. Moreover, to confirm a validity of the proposed FCDD algorithm, the hardware architecture is designed targeting on the FCDD algorithm. Synthesis results show that the hardware cost is about 1.8K gates and achieve the scalable working clock frequency in the case of FPGA (CycloneV) implementation.

(2) 3D extension

We develop a complexity reduction scheme for 3D-HEVC encoder. The proposed algorithms use fast intra texture and depth coding. Our scheme utilizes the boundary homogeneity to predict the CU sizes of the CTUs of texture and depth coding. To realize the low complexity of CU size decision, our approach notice the boundary homogeneity of every CU size. Moreover, for complexity reduction in the intra depth prediction mode, we proposed the edge classification by using Laplacian filter. The performance of the proposed algorithm was tested on a representative set of video sequences and was compared against the unmodified HTM encoder as well as two of the art complexity reduction schemes and combinations. Performance evaluations show that our proposed algorithms reduce encoding time on average 56.9% and increases BD-rate about 0.5%, compared with HTM 16.0.

(3) Future extension

We proposed the preprocessing AI software architecture for HEVC and the future coding technique. The new encoding model reduce the computation complexity significantly, and the CNN structure is simple. Specifically, CNN investigates the textures of a CU, and then determines the optimal CU/PU configuration. As compared to the reference software HM16.7, the conservative configuration of our algorithm saved 66.7% encoding time and 70.1% complexity reduction, at the cost of 1.8% BD-rate increment.

Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Background

Recently, high-definition (4Kx2K, 8Kx4K) 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 published by ISO/IEC. After the standard released, the relevant standard further work is still ongoing. The existing work of JCT-VC mainly focus on the extension of HEVC: Scalabe HEVC (SHVC) [3], 3D-HEVC [4] and HEVC Screen Content Coding (SCC) [5].

Similar to the previous video compression standards of ITU-T and ISO/IEC

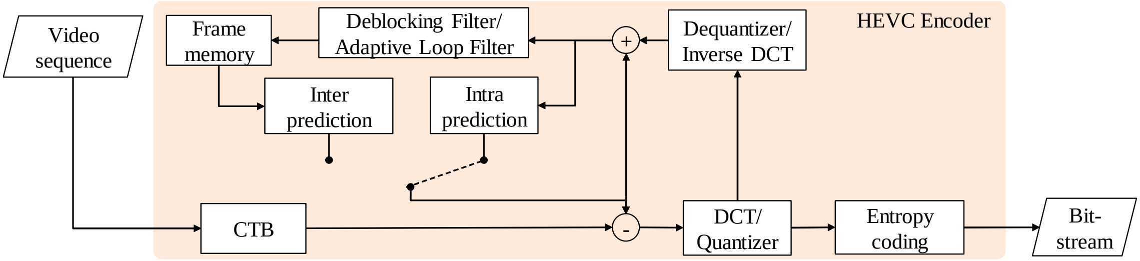


Figure 1.1 HEVC hybrid video coding structure diagram

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 figure 1.1 (CTB: coding tree block; DCT: discrete cosine transform; Q: quantization). The hybrid structures include: intra prediction, inter prediction, transform, quantization, in-loop filtering and entropy coding. Intra prediction 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 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 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 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 module is original video compressed code stream. Context-based adaptive binary arithmetic coding (CABAC) technology is used in H.265/HEVC.

1.2 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 HEVC Test Model (HM), 3D-HEVC Test Model (HTM), HEVC Test Model (SHM) are introduced briefly. In chapter 3, the two extension models of HEVC are introduced. These two extension models are important as application field of image processing, and we represent these challenge and solution. In chapter 4, a future extension model using artificial intelligence is proposed for the next video coding. In chapter 5, conclusions and the future work are introduced.

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

- CTU partitioning

Picture are divided into a sequence of coding tree unit (CTUs) [1]. The CTU concept is broadly analogous to that of the macro-block in H.264/AVC. For a picture that has three sample arrays, a CTU consists of an NxN 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 64x64. 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.

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 quad-tree 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 8x8. In a CTU, the split flag (split cu flag) is used to indicated whether CU is split into for equally-sized CU.

CU partitioning and the coding order is referred to as z-scan and is illustrated in figure 2.1. Assume that the size of coding unit CU_d is 2Nx2N, 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 NxN, and the depth is d + 1. Compared with H.264/AVC, there are some advantages using the current coding

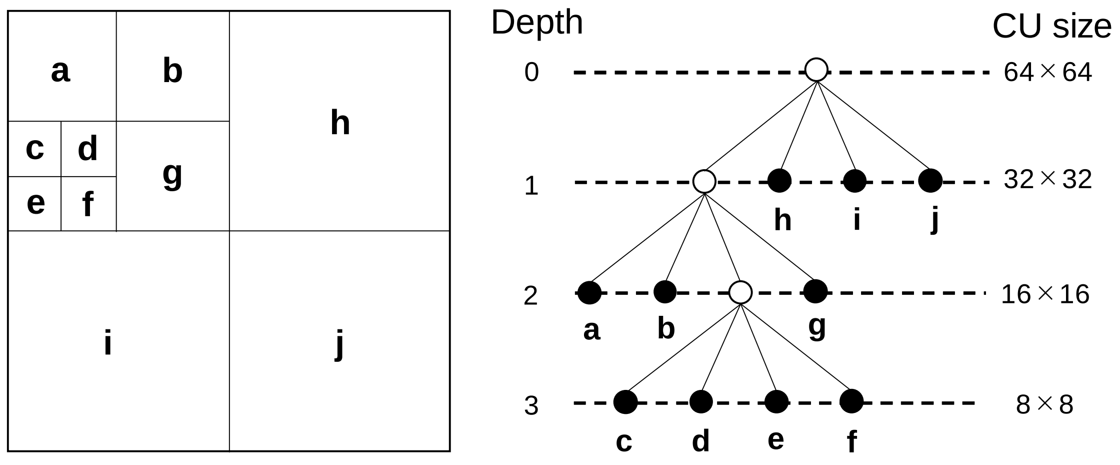


Figure 2.1 Example for the CU partitioning

structure as follow:

(1). The size of coding unit in H.265/HEVC is bigger than the macro-block in H.264/AVC. For the smooth region, it can reduce the number of bit-rate 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

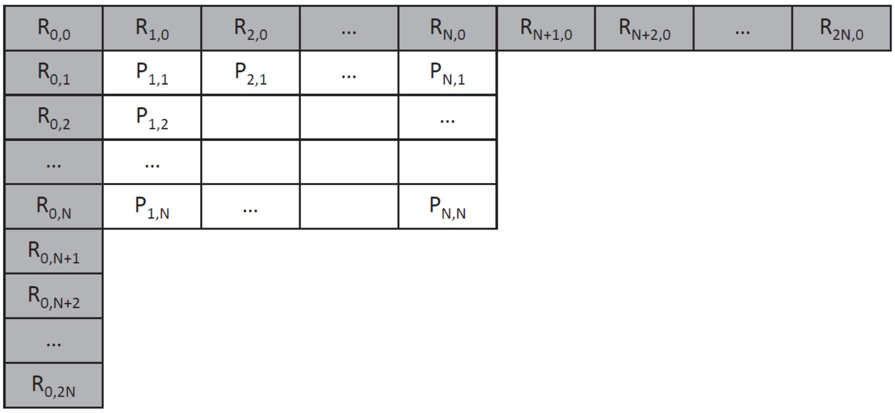
The prediction unit (PU) is a region, defined by partitioning the CU, on which the same prediction is applied. In general, the PU is not restricted to being square in shape, in order to facilitate partitioning which matches the boundaries of real objects in the picture. All prediction mode is defined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

- Transform unit (TU) structures

The transform unit (TU) is a square region, defined by quad-tree partitioning of the CU, which shares the same transform and quatization process. The TU shape is always square and it may take a size from 32x32 down to 4x4 sample. For an inter CU, the TU can be larger than CU, and it may contain PU boundaries. However, the TU can not cross PU boundaries for an intra CU. For a CU with 2Nx2N 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: 4x4, 8x8, 16x16, 32x32 and 64x64, and each PU has 35 prediction modes which contain planar mode, DC mode and 33 angular modes. The prediction template is shown as figure 2.2, 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.



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

Figure 2.2 The intra prediction template

All HEVC intra prediction modes are defined by prediction mode number as follow: planar (mode number == 0), DC (mode number == 1) and angular modes (mode numbers == 2-34). The prediction directions of the 33 angle mode is shown as figure 2.3. The direction of the mode number 2-17 mean horizontal modes, and the direction of the mode number 18-34 mean vertical modes [6].

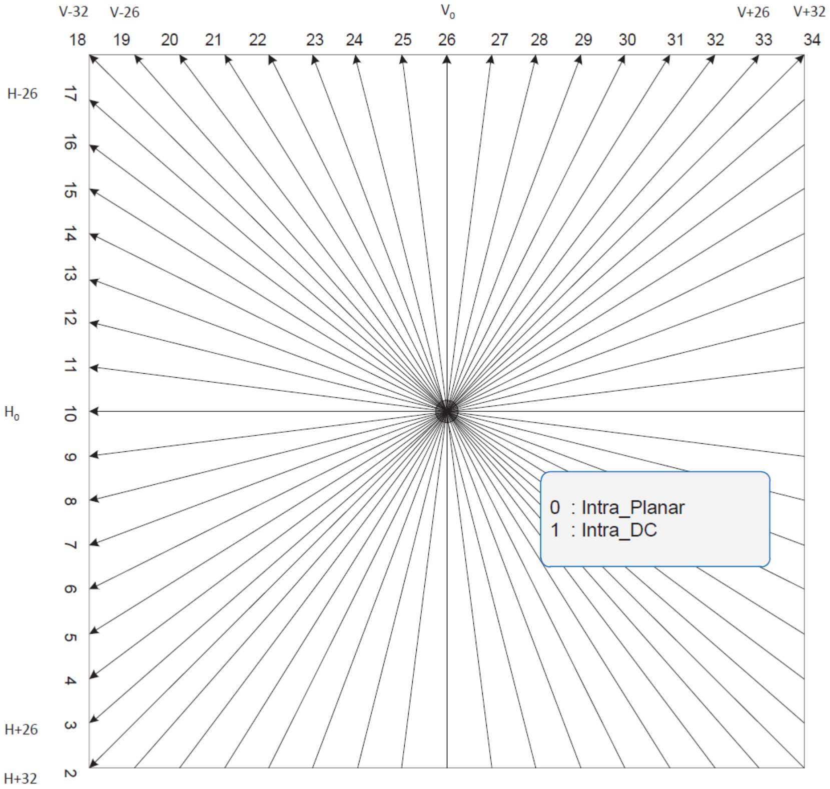


Figure 2.3 The prediction directions of the 33 angle mode

- 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 figure 2.2.

- Angular modes

There are eight different prediction directions in H.264/AVC. However, in order to adapt to the different texture of the video content, H.265/HEVC specifies 33 angular prediction modes in figure 2.3. V_0 and H_0 represent the vertical and horizontal directions, and the prediction directions of other modes 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 [7].

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

2.1.4 Transform and quantization

- Discrete cosine transform

HEVC specifies two-dimensional transforms of various sizes 4x4, 8x8, 16x16, and 32x32 that are find precision approximations to the discrete cosine transform (DCT). Multiple transform sizes improve compression performance, but also increase the implementation complexity [8]. The N transform coefficients v_i of an N-point 1D-DCT applied to the input samples ui can be expressed as

$$v_i = \sum_{j=0}^{N-1} u_j c_{ij} \tag{2.1}$$

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

$$c_{ij} = \frac{P}{\sqrt{N}} \cos[\frac{\pi}{N}(j + \frac{1}{2})i] \tag{2.2}$$

where i, $j = 0, ...N-1$ and where P is equal to 1 and $\sqrt{2}$ for $i = 0$ and $i > 0$, respectively. 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 operation. (6). It can be utilized to implement fast algorithm. For slowly changing gray 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:

$$Qstep(QP) = (2^{\frac{1}{6}})^{QP-4} \tag{2.3}$$

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 frequency dependent scaling. Human visual system based quantization can achieve better quaintly 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 [9]: a deblocking filter and a sample adaptive offset (SAO) filter. The deblocking filter aims to reduce the visibility of blocking artifacts 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 filtered. Filtering is applied to 8x8 block boundaries which are determined to be filtered, both luma and chroma components. 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 decision 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 bit-stream. SAO operation includes edge offset (OE) which uses edge properties 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 algorithmic coding (CABAC) [10]. 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 experiment for coding tool evaluations are performed. For all-intra configuration, each picture is encoded as an I frame. 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 2560x1600, 1920x1080, 832x480, 416x240, and 1280x720 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.

Chapter 3 Extension model of HEVC

3.1 Scalable extension of HEVC

3.1.1 Research motivation and all-intra coding in SHVC

The scalable extension of the high efficiency video coding (HEVC), known as SHVC, has been finalized by the joint collaborative team on video coding (JCT-VC) of the ISO/IEC MPEG and the ITU-T VCEG [13][14][15]. Taking into account the fact that the complexity of HEVC codec is higher than other existing standard codecs, as its scalable extension, simpler algorithms are expected by SHVC.

The resolution diversity of current display devices motivates the requirement for spatial scalability in SHVC. The spatial scalability is achieved by introducing multiple display resolutions within a single bit-stream. The information of the input sequences and the selected modes in the base layer (BL) can be used to estimate the optimal mode in the enhancement layer (EL), which is called inter layer reference prediction mode (ILRPM) [16]. However, ILRPM as well as intra coding in HEVC have to perform multiple times rate-distortion optimization (RDO) process, by which very high computational complexity is induced. Therefore, to construct a real-time hardware implementation, low complexity algorithm is highly required.

Many excellent works concerning the complexity reduction of HEVC are proposed from the viewpoint of candidate mode selection for intra prediction mode (IPM) and early determination of CU depth. For the computational complexity reduction of IPM, a previous work selects the candidate modes in the rough mode decision (RMD) and RDO by generating gradient-mode histogram from input pixels [17]. Similarly, an intra prediction mode decision algorithm using matching edge detector and kernel density estimation is proposed in previous work [18]. This approach reduced the computational complexity up to 25.21% with a bit-rate increase of 1.31%. However, these proposal methods are not concerning CU depth decision to archive real-time applications. As for early CU depth decision method, a fast coding unit depth decision algorithm has been proposed for the intra coding of HEVC based on the selected depth of the spatially adjacent CUs [19]. They achieved about 21% average reduction of computational complexity with 1.74% average increase in the bit-rate. From the results of [19], it was clear that the neighboring information of the encoded CUs could be used for the reduction of RDO process. Additionally, the complexity reduction of the intra coding in HEVC has been achieved by performing efficient filtering such as a pre-processing Sobel filter recently [20]. This method shows that CU depth can be estimated by detecting the edge with Sobel filter. However, in this work, a hardware implementation of the pre-processing Sobel filter for HEVC is not mentioned. Considering hardware implementation cost of pre-processing for HEVC, the hardware area might be increased when using the method of [20] because the filtering process is performed for all of pixels. To resolve the problem, an efficient hardware design of Sobel filter is proposed to reduce the hardware resources consumption of the 2D-convolution filter [21]. The developed hardware architecture achieved the reduction of the hardware resource consumption up to 98% compared to the conventional convolution implementations. However, the proposal architecture was implemented with the process of a frame-level. In this work, the hardware architecture is considered as an over-specification for a typical HEVC hardware encoder when embedding it to the HEVC encoder as a pre-processing module. In a typical HEVC hardware implementation, each function modules is highly pipelined by a coding tree unit (CTU) level because HEVC is based on a CTU structure. Another previous work noticed the problem and proposed an efficient pre-processing hardware architecture on CTU-level for real-time HEVC encoder [22]. The synthesis results by ASIC technology show 235MHz maximum clock rate and 1659 logic gate counts. The proposed architecture includes a large logic counts because the pre-processing architecture requires huge calculation using all input pixels for the edge detection. Generally, a large logic counts increase the logic critical path length. If the high working clock frequency is required for HEVC hardware encoder, the proposed hardware architecture is difficult to achieve a complete pipeline processing. Therefore, in this work, we consider that an efficient CTU-level pre-processing specification with reduced input pixels will lead to significant reduction of the hardware resource consumption.

As regards the HEVC hardware encoder specification, many previous hardware implementations achieved real-time encoding using efficient mode selection and depth estimation algorithms by CTU-level design [23][24][25] An intra encoder with source texture based CU mode pre-decision was proposed by [23]. The CU candidates are selected a rate-distortion cost (RD_cost) estimation method from the image texture. By the proposed approach, the data dependency of CTU-level in HEVC hardware was reduced. This HEVC encoder for 1080p@44fps is implemented with 2269k gate at 357MHz working clock frequency. Similarly, another work proposed a source signal based fast RMD algorithm to parallelize the hardware implementation on CTU level [24]. The design is implemented with 1571.7k gate and achieve the 1080p@60fps real-time processing at 294MHz working clock frequency. In another work, a single HEVC 8K (8,192x4,320) encoder chip has been implemented [25]. The CU depth with 8x8 and PU depth with 4x4 are eliminated for high-resolution applications. The fully parallelized VLSI architecture in which the CU depth with 64x64, 32x32, and 16x16 are parallelized was adopted to meet the 8192x4320@30fps real-time processing with 312MHz working clock frequency. In these previous work, to achieve high throughput, pipelined architecture and parallelized design are implemented. However, the hardware implementation of [25] induced the coding performance loss of 15.7% because of not performing the encoding of small CU and PU size. As a result of the discussion, not only low complexity hardware but also an efficient hardware-oriented algorithm that can achieve high-coding performance is highly required for the SHVC hardware encoder, especially for high-resolution application. Moreover, to achieve high throughput, the pre-processing hardware specification has to support high working clock frequency.

After discussing the previous works for HEVC, several excellent hardware implementations for SHVC are also noticed. Considering the software implementation of SHVC, fast software improvements are developed for those applications without considering a real-time encoding requirement. An adaptive search range method is proposed for inter coding [26]. This work reduced the coding complexity of with spatial scalability in SHVC by up to 30.27%. However, this work does not mention

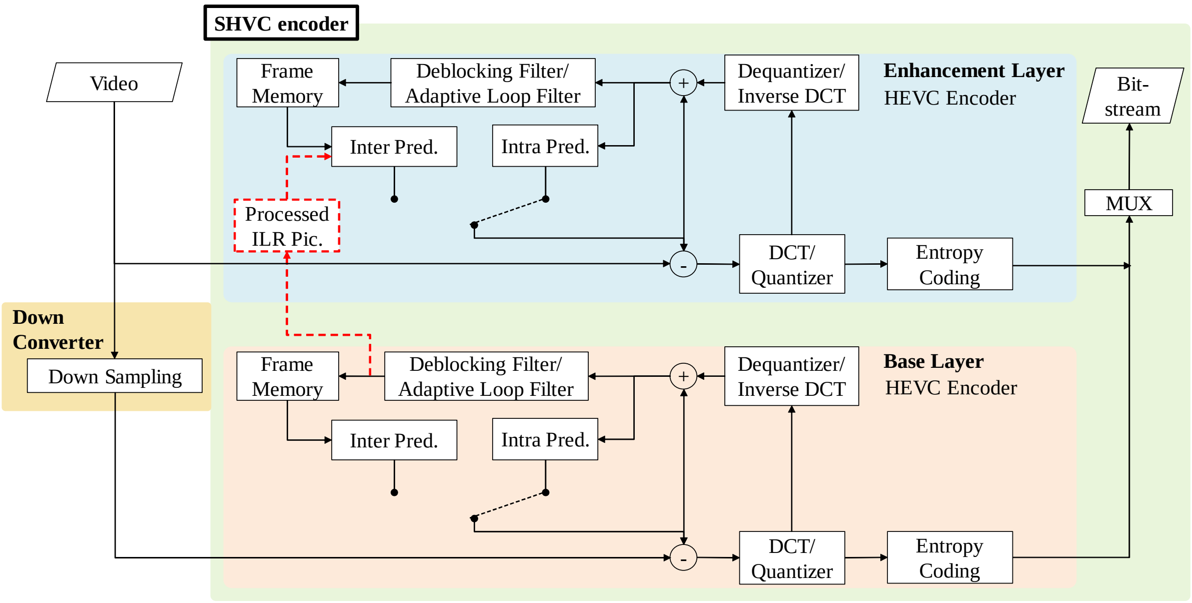


Figure 3.1 Overview of SHVC structure

the intra coding. As for the complexity reduction of intra coding, a CU depth early skip algorithm and a fast intra prediction mode decision algorithm for all intra spatial scalability is proposed [27]. In a highly related work, a fast CU splitting and pruning decision algorithm are proposed according to bayes-decision rule for SHVC [28]. This work reduced average computational complexity up to 51% and bit-rate increase of 0.79% with good coding performance. This work achieved best performance and can be considered as the state-of-the-art methodology for the complexity reduction of spatial scalability in SHVC because it can also keep high coding performance. However, this approach decided the CTU partitioning structure by selecting the patterned CU split from a probabilistic approach and the bayesian classifier using neighboring blocks of the current to-be-encoded CTU in the EL and CTU at the co-located position in the BL. Therefore, the CTU partitioning structure has to be prepared for every performing RDO process and stored in the memory in software process. Considering the hardware architecture, this approach may increase the hardware cost and the memory access frequency. Accordingly, to achieve low cost and parallel-processing hardware architecture, a CU depth decision approach that can be separated from the RDO process is considered efficient.

In our previous work, two complexity reduction algorithms were implemented to the software of SHVC (SHM11.0) [29]. Firstly, efficient CU depth determination algorithm by using ILRPM was proposed. It is clear that the selection of ILRPM or IPM has high relation with the size of CU. Secondly, fast IPM decision algorithm by using the neighboring prediction mode and an early termination process using the approximate value of the RD_cost algorithm were proposed. These fast intra coding methods were designed to SHM11.0, and the simulation result showed that the proposed algorithm achieved a low complexity calculation. However, from the viewpoint of the real-time application, our previous work did not achieve the recursive RDO process reduction for determining the best CU size. Considering the hardware implementation, the recursive RDO process involve an increase of the working clock frequency and hardware cost. Therefore, to achieve the recursive RDO process reduction, a hardware oriented optimal CU size decision method that separated from RDO process is required. Moreover, by implementing the optimal CU size decision with a simple algorithm, the hardware architecture that makes it possible to work on a low clock frequency and low cost is required. In the next section, the intra encoding in the reference software of SHVC is reviewed and the computational redundancy point will be analyzed.

3.1.2 Overview of SHVC

The SHVC architecture design enables SHVC implementations to be built using multiple repurposed single-layer HEVC codec cores, with the addition of inter layer reference prediction modules [14]. The general multi-layer high-level syntax design common to all multi-layer HEVC extensions including SHVC is represented in [15]. The video sequences of different resolution which are generated by down-sampling in DownConvert program are input to SHVC encoder. Our objective is to develop a complexity reduction method specifically designed for spatial scalability. Therefore, a novel CU depth decision algorithm with down-sampling process is proposed.

The HEVC standard inherits the well-known blockbased hybrid coding architecture of H.264/AVC [13]. However, in contrast to 16x16 pixels macro blocks (MB) used in H.264/AVC, it employs a flexible quad-tree coding block partitioning structure that enables the usage of large and multiple sizes of coding unit (CU), prediction unit (PU) and transform unit (TU). One of the frames is divided into a sequence of CTUs and the maximum size allowed for the luma block in a CTU is specified to be 64x64. Each 2Nx2N CUs which shares the same prediction mode can be divided into four smaller NxN CUs recursively until the maximum CU depth is reached. The sizes of CU range from 64x64 to 8x8. Each CU is partitioned into PUs, which is the basic unit for prediction and shares the same prediction information. PU should always be square partition with 2Nx2N or NxN (when CU is 8x8) in intra prediction. Therefore, four levels of PU search are used to find the best PU size in intra prediction, ranging from 64x64 to 4x4. The 5 sizes from 64x64 to 4x4 of PU is defined as different depths respectively, ranging from 0 to 4. Furthermore, the number of intra prediction modes (IPMs) for each PU is also increased to 35. The number of IPM for each PU is also increased to 35 compared to 9 used in H.264/AVC. The mode number of 0 is Planer and 1 is DC. The other mode number is for directional modes. As a result, the encoding complexity of intra prediction in HEVC is much more complicated than H.264/AVC.

SHVC has been modified in such a way that the collocated reconstructed pictures from the reference layers can be used as inter layer reference prediction when coding the current EL. The red dotted line as shown in figure 3.1 is used to generate the inter layer reference prediction. The spatial scalability is a scalability function that has been adopted in SHM11.0. The ILRPM enables the prediction between different resolutions in order to achieve this function in SHVC. Moreover, improvement of the coding efficiency is realized by using the texture information in BL. Accordingly, ILRPM requires the encoded information of the BL, and need to calculate the RD_cost by all CU depth. Therefore, the computational complexity for the RDO calculation is increased compared with HEVC.

All-intra mode of the EL in SHM11.0 is performed as the following process. Firstly, IPM of the same 35 modes as HEVC is performed for EL. Next, ILRPM is performed using the upsampled texture information at the co-located position in the BL. Finally, the optimal mode of the EL which is intra EL or ILRPM with minimum RD_cost is determined. Thus, because the ILRPM is also performed by each CU depth, the computational complexity in EL is particularly large. In the next section, for the complexity reduction of CU depth decision, we introduce a parameter named boundary correlation.

3.1.3 Proposed algorithm

3.1.3.1 Fast CU depth decision by boundary correlation

As shown in the previous section, the recursive RDO calculation of SHVC encoder is required to determine the optimal CU depth. Therefore, to reduce the process of determining the best CU depth, the proposed algorithm for fast CU depth decision by using boundary correlation (FCDD) is represented.

To clarify the boundary correlation of the CTU area, our approach uses the boundary pixels in a CU from 8x8 to 64x64 blocks which are shown in figure 3.2. In the case an 8x8 CU is selected, a parameter l is set to 0. The pixels which are used to calculate the boundary correlation are shown in figure 3.2 (a). Similarly, the boundary pixels from 16x16 to 64x64 block are represented by (b), (c), and (d). The average value of boundary correlation (bc) in each CUs is calculated by

$$\begin{aligned} & bc_1(k,m) \\ & = \sum_{j=0}^{2^{3+l}-1} P(x_{2^{2+l}-1+2^{3+l}(k-1)}, y_{j+2^{3+l}(m-1)}) >> 3+l \end{aligned} \tag{3.1}$$

$$\begin{aligned} & bc_2(k,m) \\ & = \sum_{j=0}^{2^{3+l}-1} P(x_{2^{2+l}+2^{3+l}(k-1)}, y_{j+2^{3+l}(m-1)}) >> 3+l \end{aligned} \tag{3.2}$$

$$\begin{aligned} & bc_3(k,m) \\ & = \sum_{j=0}^{2^{3+l}-1} P(x_{j+2^{3+l}(k-1)}, y_{2^{2+l}-1+2^{3+l}(m-1)}) >> 3+l \end{aligned} \tag{3.3}$$

$$\begin{aligned} & bc_4(k,m) \\ & = \sum_{j=0}^{2^{3+l}-1} P(x_{j+2^{3+l}(k-1)}, y_{2^{2+l}+2^{3+l}(m-1)}) >> 3+l \end{aligned} \tag{3.4}$$

where $P(x,y)$ denotes the pixel position in a CTU. k and m represent the number of divided CUs. In this work, when calculating the boundary correlation for 8x8, 16x16, 32x32, and 64x64, the maximum number of k and m are equal to 8, 4, 2, and

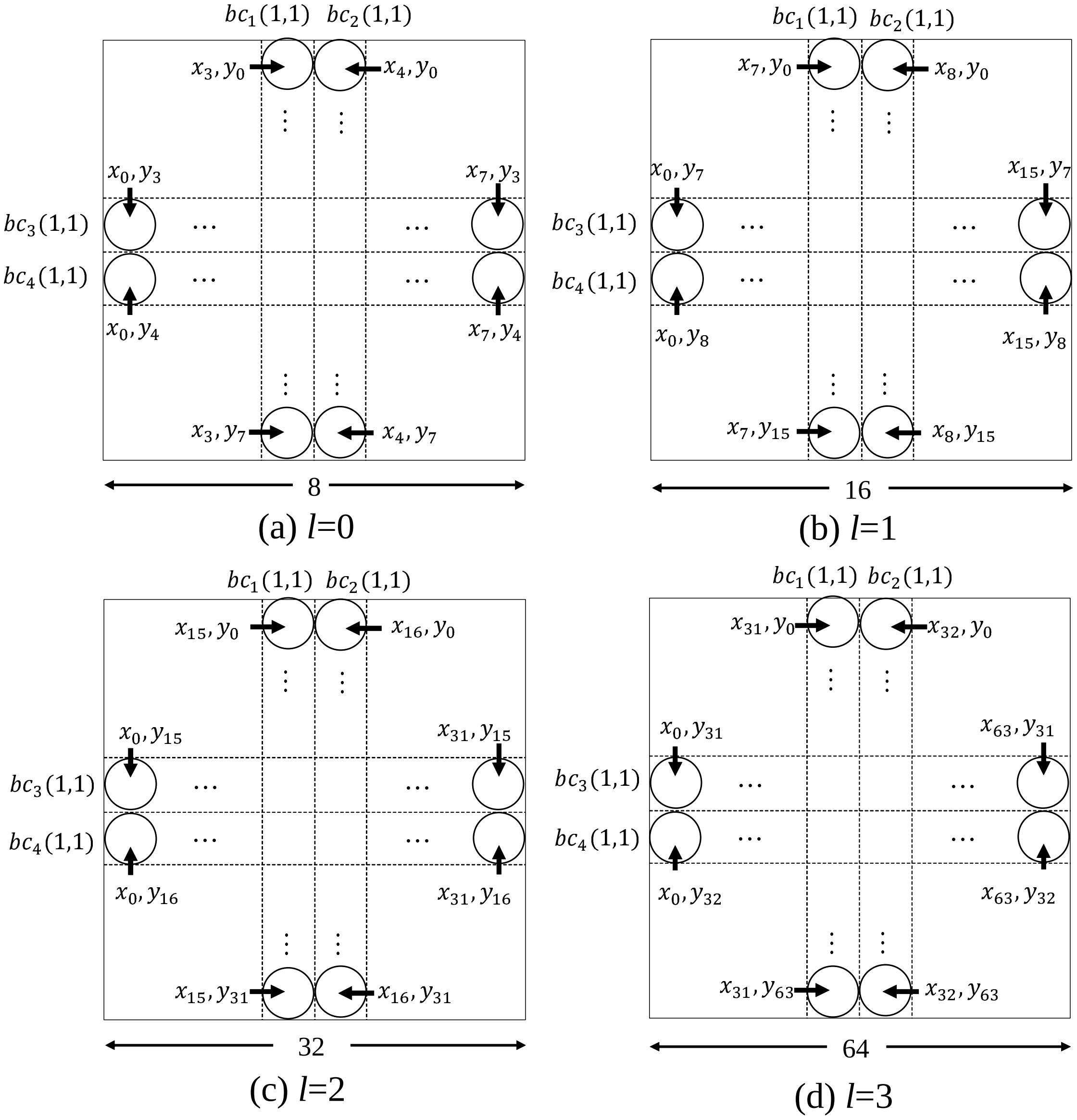


Figure 3.2 Overview for calculating the boundary correlation in the case of $k = 1$ and $m = 1$

1, respectively. The horizontal and vertical difference value of $bc_1(k,m)$, $bc_2(k,m)$, $bc_3(k,m)$, and $bc_4(k,m)$ are defined as

$$bc_V(k,m) = |bc_1(k,m) - bc_2(k,m)| \tag{3.5}$$

$$bc_H(k,m) = |bc_3(k,m) - bc_4(k,m)| \tag{3.6}$$

where $bc_V(k,m)$ and $bc_H(k,m)$ indicate the homogeneous relation for each CU using the pixel values of vertical line and horizontal line in a CU boundary. These values realize the optimal CU depth selection with few pixel information and simple calculation.

The boundary strength correlation is represented by $bc_V(k,m)$ and $bc_H(k,m)$ of the previous section. When $bc_V(k,m)$ or $bc_H(k,m)$ are small, the boundary correlation is high. Because the reference pixels for IPM use the neighboring pixel, the reference pixels for IPM can be easily predicted when $bc_V(k,m)$ or $bc_H(k,m)$ are small. In other words, the optimal CU size can be encoded by large CU size. However, the threshold (TH) value using $bc_V(k,m)$ and $bc_H(k,m)$ depend on QP because the optimal CU depth selection in SHVC is determined by RDO [31]. Therefore, in our proposed FCDD algorithm, as summarized in Table 3.1, TH is adaptively selected by different QP value.

Table 3.1 Boundary threshold values according to QP

Condition	TH
QP $\leq$ 27	10
27 $<$ QP $\leq$ 32	20
32 $<$ QP $\leq$ 37	30
37 $<$ QP	50

In summary, the proposed FCDD algorithm with analyzing the block from 8x8 pixels to 64x64 pixels is represented in Algorithm 1. Firstly, the number of the process in FCDD is calculated from X using l (lines 1-2). The boundary correlation of the current CU size are calculated from $bc_1(k,m)$, $bc_2(k,m)$, $bc_3(k,m)$, and $bc_4(k,m)$ (line 5). $bc_V(k,m)$ and $bc_H(k,m)$ are calculated by in $bc_1(k,m)$, $bc_2(k,m)$, $bc_3(k,m)$, and $bc_4(k,m)$, and BC is defined as max value of $bc_V(k,m)$ and $bc_H(k,m)$ (lines 6-8). In the next stage, BC is compared with TH (line 9). If BC is larger than TH, the CU size of 2^{2+l}x2^{2+l} is determined as the optimal CU size (line 10). For example, when l is equal to 0, all of the boundary correlation of 8x8 pixels in a CTU is evaluated (lines 3-13). Similarly, the block from 16x16 pixels to 64x64 pixels are performed with the same process.

Algorithm 1 : Fast CU depth decision algorithm

```
1:  for $l = 0$ to 3 do
2:      $X \leftarrow 64/2^{3+l}$                    ▷ The number of boundary calculation
3:      for $m = 1$ to $X$ do        ▷ Evaluate the boundary pixel of every CU size
4:          for $k = 1$ to $X$ do
5:              Calculate $bc_1(k,m), bc_2(k,m), bc_3(k,m)$ and $bc_4(k,m)$.
6:              $bc_V(k,m) \leftarrow |bc_1(k,m) - bc_2(k,m)|$
7:              $bc_H(k,m) \leftarrow |bc_3(k,m) - bc_4(k,m)|$
8:              $BC \leftarrow max\{bc_V(k,m), bc_H(k,m)\}$
9:              if $BC \geq$ TH then
10:                 OptCUsize $\leftarrow 2^{2+l}$x$2^{2+l}$        ▷ Determine the optimal CU size
11:             end if
12:         end for        ▷ Evaluate the next horizontal line into CTU and goto 4
13:     end for              ▷ Evaluate the next vertical line into CTU and goto 3
14: end for                                          ▷ Evaluate the next CTU goto 1
```

3.1.3.2 Analysis of spatial layer relationship

To realize real-time application in SHVC, significantly computational complexity reduction is required by using the information in the BL. Therefore, it is an important issue to efficiently use the available information in the BL.

ILRPM is a very efficient mode for all-intra spatial scalability. However, according to the characteristics of a video sequence, there are still some CUs coded by IPM in EL instead of ILRPM. Based on the reference software [15], we evaluate the probability that the ILRPM is used as the best mode in order to confirm the importance of ILRPM. Test sequences from Class 4K to Class E in Table 3.5 are tested with the quantization parameters (QPs) set to 22, 27, 32, 37 when the spatial scalable ratio is 2 for the spatial scalability. Table 3.2 shows the activity ratio of the ILRPM with the average value in the set of each QPs. This probability (P) is defined as

Table 3.2 Activity ratio of the ILRPM (%)

Test sequnces	Resolution	P by CU depth			
		64x64	32x32	16x16	8x8
Class 4K	3840x2160	90.21	89.00	85.15	78.27
Class A	2560x1600	91.11	90.37	83.35	80.94
Class B	1920x1080	90.16	89.72	86.00	70.83
Class C	832x480	96.63	93.30	88.92	79.81
Class D	416x240	96.85	91.01	87.78	82.14
Class E	1280x720	93.31	87.87	84.15	75.37
Average		93.45	90.21	85.89	77.89

$$P = \frac{Num_{ILRPM}}{Num_{ALL}} \times 100\,(\%) \tag{3.7}$$

The number of the CU that selected ILRPM in one sequence is defined as Num_{ILRPM}. All the number of CU in the sequence is defined as Num_{ALL}.

From Table 3.2, it is clear that for large CUs ILRPM is selected with very high probability. On the other hand, in the CUs encoded by 8x8, only 77% of CUs are coded by ILRPM. In other words, about 23% are encoded by IPM. It is clear that the selection of ILRPM or IPM has high relation with the size of CU. As well known, when a CU tends to be smooth with less high frequency components, generally a larger CU will be selected. In this case, ILRPM works well and over 90% of large CU such as 64x64 and 32x32 are encoded by ILRPM. Therefore, for large CUs, IPM can be reduced from the candidate modes. On the other hand, if a small CU is selected it has more high frequency components. Therefore, if a smaller CU is selected, more high frequency loss occurred during the down-sampling process and more noise included during the up-sampling process. It is the reason that ILRPM which including a up-sampling process is decreasingly selected when the size of CU decreased. It is also the reason that IPM makes an important role for small CUs. Accordingly, not only ILRPM but also IPM are necessary for the encoding process of EL when small CU such as 16x16 and 8x8 are selected.

The evaluation results lead to the conclusion that the checking order of ILRPM should be with high priority. In this work, ILRPM is firstly checked and large CUs with 64x64 and 32x32 are not used as the candidates for IPM. Thereby, the proposed method can reduce not only the computational complexity of RDO process in EL but also the number of processing cycles for hardware implementation.

3.1.3.3 Evaluation of IPM and RD_cost

In section 3.1.3.1, the CU depth is determined by boundary correlation before encoding in our scheme. However, the computational complexity for SHVC include not only the CU depth decision but also the candidate mode decision in recursive RDO process. Therefore, it is difficult to realize the real-time application by using only fast CU depth decision approach. In this section, to reduce the calculation of the candidate mode decision in recursive RDO process, the candidate mode and the RD_cost are evaluated for fast coding.

From the verification in the previous section, more IPM is used with small PU size of 16x16, 8x8, and 4x4 because the directional mode can generate accurate prediction candidate to high resolution sequences. However, as shown in section 3.1.2, the optimal mode decision process using the candidate modes in RMD and RDO induce high computational complexity. Therefore, the reduction of the candidate modes in RMD and RDO can help to reduce the computational complexity of IPM.

Figure 3.3 shows an illustration of the mode number in the neighboring IPM ($\text{IPM}_{Neighbor}$) and the co-located position in the BL (IPM_{BL}). In the related work [30], most probable modes (MPMs) show the high accuracy as the prediction candidate. From the related work, $\text{IPM}_{Neighbor}$ are considered as an efficient for the reduction of the prediction mode. Moreover, considering the spatial relationship, IPM_{BL} is also effective as a candidate mode. To prove the effectiveness of $\text{IPM}_{Neighbor}$ and IPM_{BL}, we evaluate the possibility which $\text{IPM}_{Neighbor}$ and IPM_{BL} are the same mode with mode number of best mode in the current PU (IPM_{Best}), which is illustrated in Table 3.3. Table 3.3 shows that $\text{IPM}_{Neighbor}$ and IPM_{BL} of candidate modes can cover about 90% of IPM_{Best} on average. Therefore, in our proposed algorithm, RMD

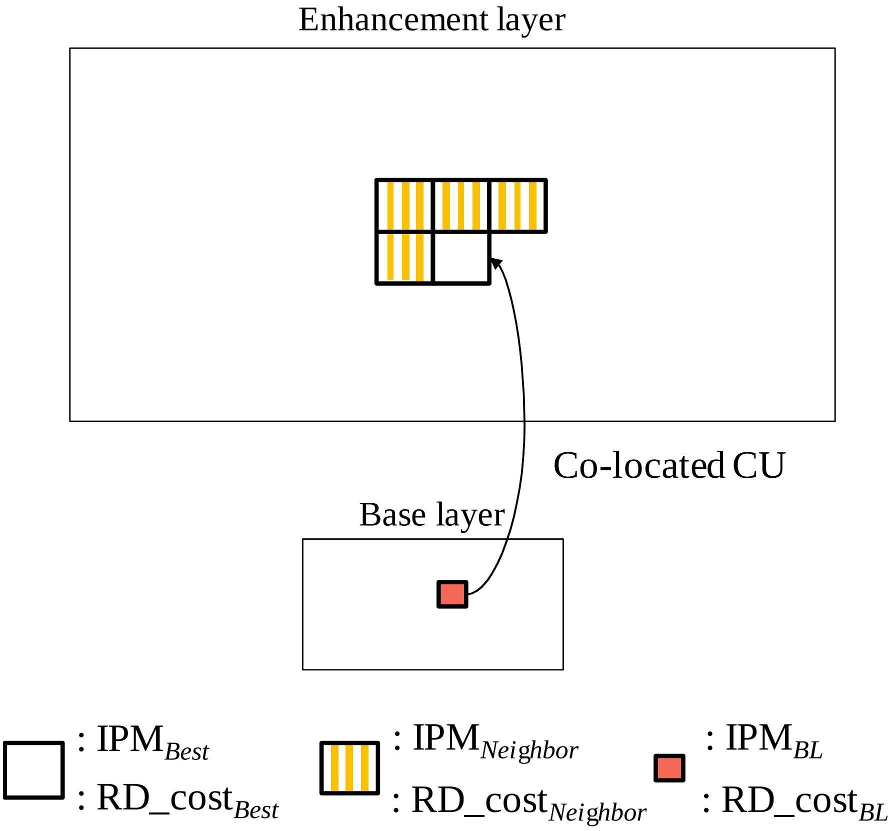


Figure 3.3 Mapping IPM and RD_cost

and RDO process are reduced by using $\text{IPM}_{Neighbor}$ and IPM_{BL}.

Table 3.3 Probability of the same mode as the IPM_{Best} (%)

PU size	Class 4K	Class A	Class B	Class C	Class D	Class E
16x16	93.64	90.21	89.25	95.42	93.65	96.10
8x8	91.45	92.01	90.05	95.11	94.42	95.00
4x4	92.12	93.33	92.98	91.29	91.88	95.61

The RD_cost calculation also include the recursive process. From our previous work [29], we found that the RD_cost of IPM in EL has no much difference with the RD_cost at the co-located position in the BL. However, in [29], when IPM in PU size with 16x16, 8x8, and 4x4 are encoded, the four neighboring blocks of the current to-be-encoded CTU in the EL ($\text{RD\_cost}_{Neighbor}$) and CTU at the co-located position in the BL (RD_cost_{BL}) are not analyzed. For this reason, we evaluate the ratio (*Cost_Ratio*) of RD_cost in the best mode (RD_cost_{Best}) and the candidate of most minimum RD_cost (RD_cost_{Min}) by a histogram. *Cost_Ratio* and RD_cost_{Min} are defined as

$$\begin{aligned} Cost\_Ratio &= \frac{\text{RD\_cost}_{Best}}{\text{RD\_cost}_{Min}} && (3.8) \\ \text{RD\_cost}_{Min} &= \min\{\text{RD\_cost}_{Neighbor}, \text{RD\_cost}_{BL}\} && (3.9) \end{aligned}$$

Since the RD_cost is sensitive to QP, *Cost_Ratio* is evaluated with various QP values [29]. Figure 3.4 shows the histogram of different QP in the sequence of Class 4K and Class A. From the evaluation result in figure 3.4, we found that *Cost_Ratio* follows a normal distribution. Therefore, by using the normal distribution, adaptively early termination condition of RDO process is possible to be proposed according to QP.

3.1.3.4 Fast mode decision for IPM

As discussed in section 3.1.3.3, when IPM in PU size with 16x16, 8x8, and 4x4 is encoded, it is clear that $IPM_{Neighbor}$ and IPM_{BL} are selected with very high probability for the best prediction mode. Generally, when IPM in PU size with 16x16, 8x8, and 4x4 is encoded, the directional mode is probably used as best prediction mode. However, the SHVC encoder has to select one best mode from a large number of candidate modes including 33 directional modes. Accordingly, we consider that the directional modes induce the redundant calculation.

To reduce the redundant calculation, high spatial correlation can be utilized for the directional mode reduction. Because our verification results show that the neighboring block and the block at the co-located position in the BL have high spatial correlation, our proposed FMD algorithm use 4 $IPM_{Neighbor}$ in the EL, IPM_{BL}, DC mode, and Planer mode as the candidate modes. In worst case, IPM of RMD process in EL is performed by using 7 modes. Our proposed FMD algorithm also reduces the candidate modes in RDO process. When our proposed FMD algorithm is applied to IPM, the method of generating prediction pixels is largely different. If the encoding block has more high frequency components, the possibility that a similar mode will be included in the candidate lists is low. Therefore, a comparison

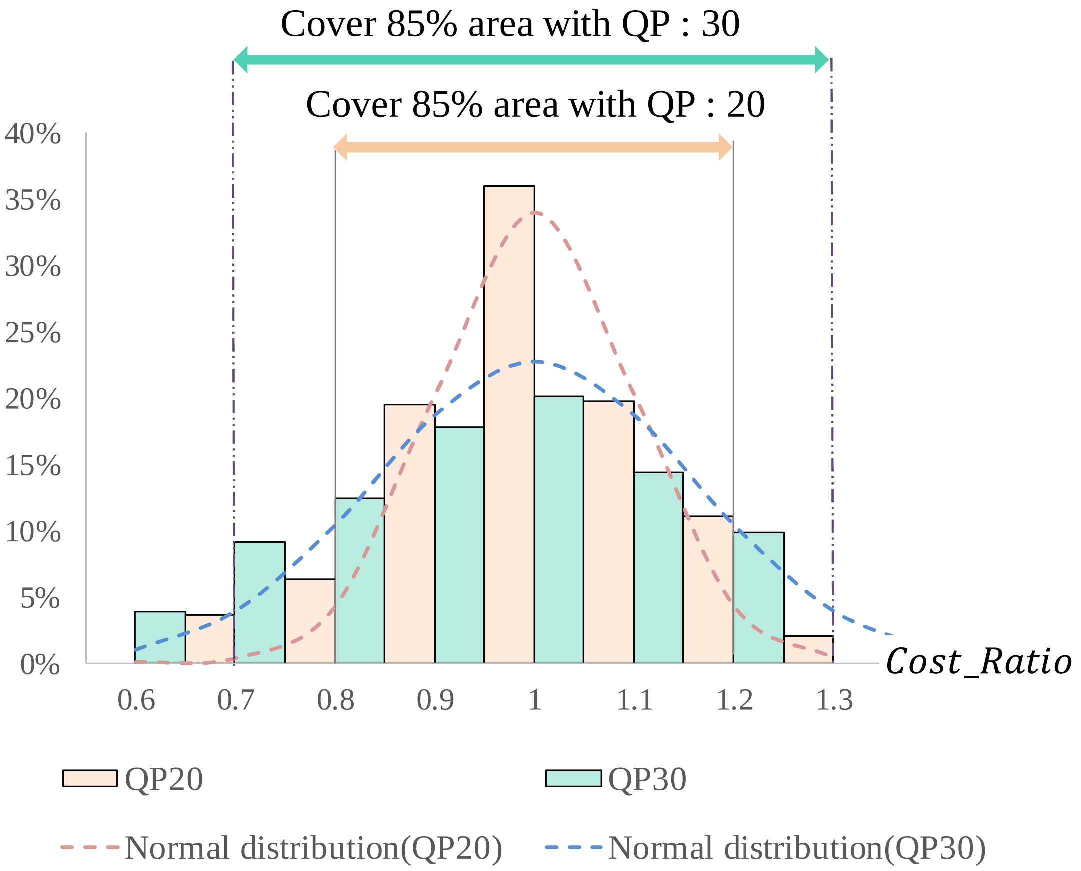


Figure 3.4 Histogram and normal distribution of *Cost_Ratio* in QP = 20 and QP = 30

of 2 most possible modes is required. Our proposed FMD algorithm select the first 2 most possible modes from the candidate lists. The detail of the reduction mode number is represented by Table 3.4.

3.1.3.5 Early termination process using the RD_cost

In the previous section, an efficient algorithm which reduces the number of prediction mode on the RMD and RDO process is proposed. However, RDO process has high complexity compared with RMD process. Therefore, the reduction of the calculation in RDO process is highly required.

By analyzing the histogram in figure 3.4, it can be noticed that the distribution

Table 3.4 Number of candidate mode for RMD and RDO process in worst case

PU size	Number of IPM in proposed algorithm		Number of IPM in SHM11.0	
	RMD	RDO	RMD	RDO
16x16	7	2	35	3
8x8	7	2	35	8
4x4	7	2	35	8

of *Cost_Ratio* follows a normal distribution. In figure 3.4, the normal distribution can be approximated by

$$f(x) = \frac{1}{11\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(x-1)^2}{2\sigma^2}\right\} \tag{3.10}$$

where x represent *Cost_Ratio* and σ is given by

$$\sigma \approx \begin{cases} 0.10 & (\mathrm{QP}=20) \\ 0.16 & (\mathrm{QP}=30) \end{cases}$$

Furthermore, it is confirmed by a previous work that when the range of *Cost_Ratio* makes the coverage rate bigger than 85%, the coding performance will not change much [29]. It can also be described as

$$\int_{1-s}^{1+s} f(x)dx = 0.85 \tag{3.11}$$

where s represents the distance from the center. In figure 3.4, the value of s can be approximated as

$$\begin{cases} s = 0.2 & (\mathrm{QP}=20) \\ s = 0.3 & (\mathrm{QP}=30) \end{cases} \tag{3.12}$$

Moreover, from simulation results under different QPs, we note that the relationship between *Cost_Ratio* and QP value can be represented by a linear approximate. Now, *Cost_Ratio* is represented from RD_Cost_{Best} and RD_Cost_{Min}. However,

RD_Cost_{Best} value is not available while encoding the current CU. Accordingly, our approach use RD_cost of current CU ($\text{RD\_Cost}_{Candidate}$) for the early termination process (ETP) in which an adaptively early termination condition is applied for RDO process. The adaptively early termination condition is represented as

$$|\frac{\text{RD\_Cost}_{Candidate}}{\text{RD\_Cost}_{Min}} - 1.0| \leq 0.01 \times \text{QP} \tag{3.13}$$

For showing the detail of ETP, the proposed FMD with ETP algorithm is summarized as Algorithm 2.

Algorithm 2 : FMD with ETP algorithm

```
PU_size ← 16x16 or 8x8 or 4x4
for each PU_size in CTU do
    $IPM_{List} \leftarrow IPM_{Neighbor}, IPM_{BL}, DC, Planer$    ▷ Input the prediction mode number
    for All $IPM_{List}$ do    ▷ RMD process
        Calculate RMD_Cost value with mode number of $IPM_{List}$
        if CurrRMD_Cost ≥ RMD_Cost then
            $RDO_{List} \leftarrow IPM_{List}$
        end if
    end for
    for $RDO_{List} < 2$ do    ▷ RDO process
        $\text{RD\_Cost}_{Min} \leftarrow \min\{\text{RD\_cost}_{Neighbor}, \text{RD\_cost}_{BL}\}$
        Calculate $\text{RD\_Cost}_{Candidate}$ value with $RDO_{List}$
        if Condition (3.13) is satisfied then
            Terminate process early, and next CU
        end if
    end for
end for
```

3.1.3.6 Overall process

The flowchart of the overall proposed algorithm is shown in figure 3.5. First, the boundary correlation in each layer (BL and EL) are analyzed by our proposed

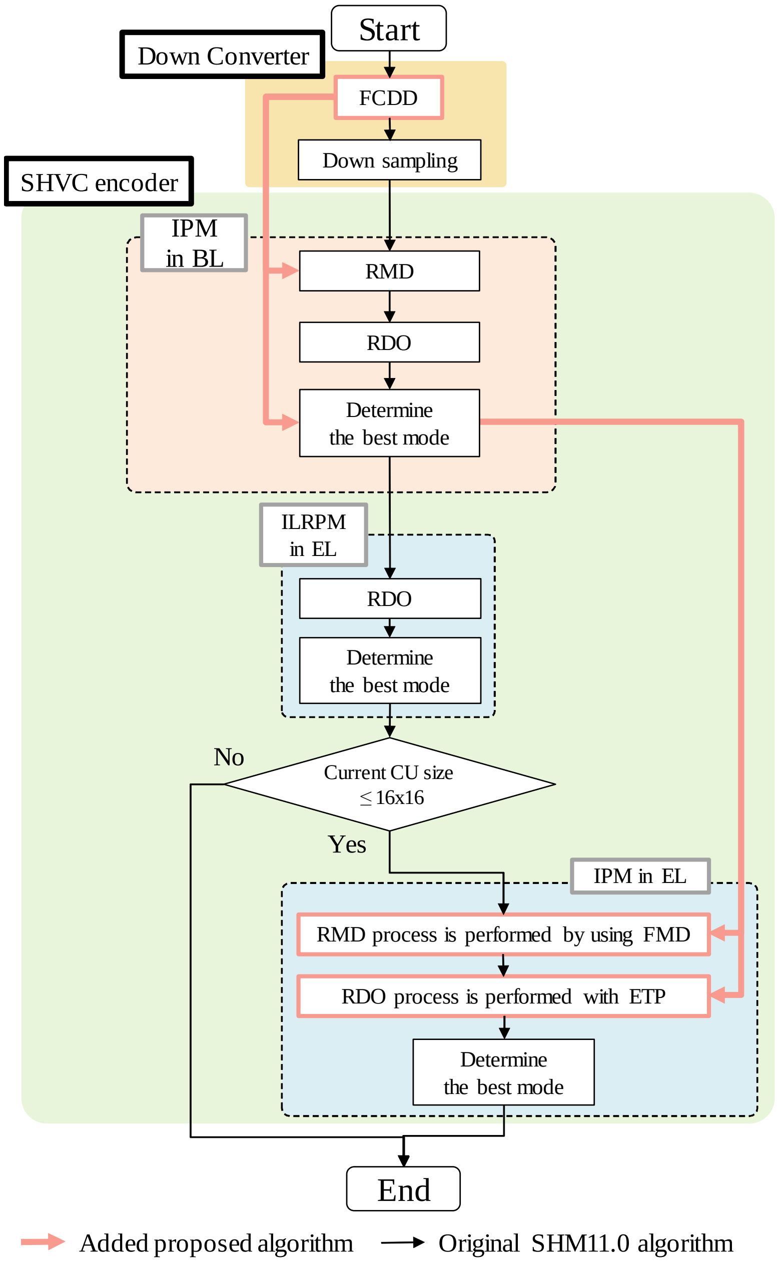
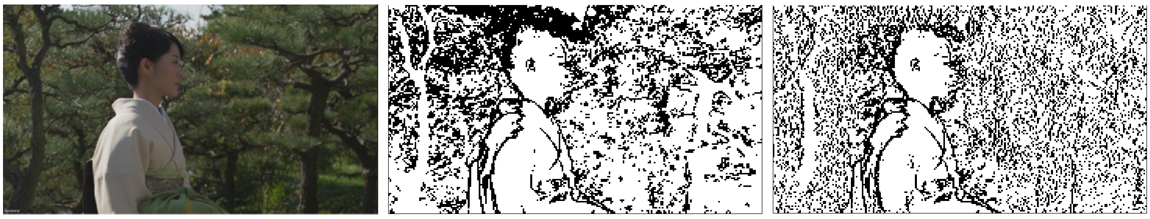


Figure 3.5 Flowchart of the overall proposed algorithm.

FCDD algorithm. Then, the down-sampling in SHM 11.0 is performed to generate the original video sequences of BL. Our proposed FCDD algorithm generates the mapping information of the optimal CU depth, and their information are input to the encoder of BL and EL. When the current CU is the block size of 64x64, 32x32, 16x16, and 8x8, ILRPM is performed because it is confirmed by the previous work that ILRPM is selected with high probability. On the other hand, IPM is performed with



(a)Input sequence (b)Sobel filter edge (c)Our proposed

Figure 3.6 Comparison of the edge detection with sobel filter and our proposed.

our proposed fast mode decision (FMD) algorithm and early termination process (ETP) when using the block size of 16x16 and 8x8. Our proposed FMD algorithm use 4 prediction modes of CU$_{Neighbor}$ (which are located at left, left upper, upper, and right upper) in the EL, a prediction mode of CU$_{BL}$, DC mode, and Planer mode as the candidate modes. In worst case, IPM of RMD process in EL is performed by using 7 modes. Moreover, in RDO process, the proposed ETP predict RD_Cost of best mode by using RD_Cost in CU$_{Neighbor}$ and CU$_{BL}$. The proposed ETP judge whether it is the early termination condition or not [29]. As the result, the best CU partitioning and prediction mode are fast determined.

3.1.4 Simulation results

In this section, to verify the coding performance of our proposed algorithm, two evaluation are represented. Firstly, the prediction accuracy of FCDD algorithm for CU depth decision and the coding performance are evaluated. Secondly, the coding performance of all proposed algorithms is evaluated with SHVC.

3.1.4.1 Evaluation of FCDD algorithm in HEVC

To evaluate the edge detection algorithm, the difference of the edge detection in the sequence of ”Kimono” is shown in figure 3.6. As shown in figure 3.6, we consider that our proposed method can detect the important edge part for encoding. To clear that this method is good for encoding, we evaluate the prediction performance by different THs using different QPs with the sequence of Class 4K and Class A. The following test dataset which cover not only Class A, B, C, D, and E but also some 4K sequences are used for subsequent verification and simulation. Table 3.5 summarizes the specifications of this test dataset [32][33][34].

Table 3.5 Configuration of encoded test sequences

Class	**Sequences**	**Frame Length**
Class 4K (EL:3840 × 2160) (BL:1920 × 1080)	Beauty Bosphorus HoneyBee Jockey ReadySetGo ShakeNDry YachtRide	50
Class A (EL:2560 × 1600) (BL:1280 × 800)	PeopleOnStreet Traffic	100
Class B (EL:1920 × 1080) (BL:960 × 540)	BasketballDrive BQTerrace Cactus Kimono ParkScene	100
Class C (EL:832 × 480) (BL:416 × 240)	BasketballDrill BasketballDrillText PartyScene RaceHorses	150
Class D (EL:416 × 240) (BL:208 × 120)	BasketballPass BQSquare BlowingBubbles	150
Class E (EL:1280 × 720) (BL:640 × 360)	FourPeople Johnny KristenAndSara SlideEditing SlideShow Vidyo1	150

Table 3.6 shows the CU depth prediction accuracy (*Acc*) for every QP when the optimal CU depth is determined by using different TH values. *Acc* was computed using the following equation

$$Acc(\%)$$

Table 3.6 *Acc* according to QP (%).

	QP = 22	QP = 27	QP = 32	QP = 37
TH = 10	83.3	83.0	55.7	45.2
TH = 20	63.2	86.1	81.2	68.5
TH = 30	55.4	68.2	82.1	79.8
TH = 50	45.3	67.5	73.2	88.6

$$= \sum_{f=1}^{F}\sum_{j=1}^{J}\frac{\frac{\sum_{i=1} SCU_{(i,j,f)}\times CorrectCU_{(i,j,f)}}{\sum_{i=1} SCU_{(i,j,f)}}}{J\times F}\times 100 \tag{3.14}$$

where $CU_{(i,j,f)}$ indicates the ith CU of the jth CTU of the fth frame of the test video [28]. $SCU_{(i,j,f)}$ represents the area (number of pixels in width × number of pixels in height) of the $CU_{(i,j,f)}$. $CorrectCU_{(i,j,f)}$ is equal to one if the CU depth of $CU_{(i,j,f)}$ is predicted correctly ($CorrectCU_{(i,j,f)}$). J and F are the total number of CTUs in each frame and the total number of frames of the test video, respectively.

As shown in Table 3.6, *Acc* is very different when QP changes. Generally, the value of the optimal TH increased when QP is increasing. It's also clear that the proposed algorithm can archive high prediction accuracy even a fixed TH value is used for the different size of CUs. This result means that TH value is very sensitive to QP, but not sensitive to CU size.

To evaluate the coding performance of our FCDD algorithm, the proposed algorithm is implemented to HEVC software (HM16.7) [35] and compared to other previous works in the sequence of Class A and Class B. The difference value of execution time between the unmodified HM16.7 and the proposed algorithm is represented as time saving (TS). TS is defined as

$$TS = \frac{T_{HM16.7} - T_{Proposed}}{T_{HM16.7}} \times 100\,(\%) \tag{3.15}$$

where $T_{HM16.7}$ is the encoding time of the unmodified HM16.7 and $T_{Proposed}$ is that of the proposed algorithm. The performance of our proposed complexity reduction is compared with that of the unmodified HM16.7, and impact on the bit-rate (bjontegaard delta rate (BD-BR)). Table 3.7 shows that FCDD algorithm reduces TS by an average of 39.8% at the cost of 0.9% BD-BR increase. BD-BR shows a good performance compared to other works. Considering hardware implementation, the FCDD algorithm can achieve low hardware resource consumption than the previous work because FCDD algorithm use only the pixels on the edge of CUs instead of all the pixels in a CTU [20].

Table 3.7 Comparison of previous works and the proposed FCDD algorithm in HEVC

	Chen's algorithm [18]		Na's algorithm [20]		Proposed	
Class	TS(%)	BD-BR(%)	TS(%)	BD-BR(%)	TS(%)	BD-BR(%)
Class A	24.0	0.9	59.9	2.2	38.8	1.0
Class B	22.7	1.0	67.5	1.6	40.9	0.8
Average	23.4	1.0	63.7	1.9	39.8	0.9

3.1.4.2 Coding performance in SHVC

For the evaluations of all proposed algorithm with reference software SHM11.0, a two-layer spatial scalability structure are used with all-intra mode configuration [36]. As suggested in common SHM test conditions, we test our proposed algorithm for spatial ratio 2. For spatial scalability, the BL is generated by down-sampling the original video streams using the DownConvert program in SHM 11.0. The performance of our proposed complexity reduction is compared with that of the unmodified SHM11.0 encoder in terms of execution time, and impact on bit-rate and peak signal to noise ratio (PSNR). The difference value of execution time between the unmodified SHM11.0 and the proposed algorithm is represented as time saving (TS). TS is defined as

$$TS = \frac{T_{SHM11.0} - T_{Proposed}}{T_{SHM11.0}} \times 100\,(\%) \tag{3.16}$$

where $T_{SHM11.0}$ is the encoding time of the unmodified SHM11.0 and $T_{Proposed}$ is that of the proposed algorithm.

The results of our experiment are summarized in Table 3.8, 3.9, Table 3.10. In Table 3.8, 3.9, the time reduction percentage compared to the unmodified SHVC encoder is shown, the impact on BD-BR, and the video quality in terms of PSNR

(BD-PSNR) [37].

For the spatial scalability, we have examined two scenarios for 1) FCDD and 2) FMD with ETP. Table 3.8 shows the BD-BR and BD-PSNR results for the spatial scalability for the all-intra mode configuration. The QPs of the BL and EL were set to (QPBL, QPEL) = {(22,22), (27,27), (32,32), (37,37)} [37]. For the spatial scalability of scalable ratio 2, we observe that if the FCDD is used, the average BD-BR value is about 0.50%. The average TS for FCDD is 39.60%.

Our proposed FMD with ETP algorithm reduces TS by an average of 45.18% at the cost of 0.68% bit-rate increase compared to the unmodified SHVC encoder. Compared with SHM11.0 using FCDD, the average TS for FMD with ETP particularly represent the high complexity reduction in Class 4K because much small PU size such as 16x16, 8x8, and 4x4 block are included to Class 4K. Small PU size needs to select the best prediction mode from a large number of candidate modes including 35 intra prediction. The mode selection process induce the redundant calculation compare with the recursive RDO process for determining the best CU size. Therefore, our proposed FMD with ETP algorithm can be effective for high-resolution such as Class 4K. Moreover, our proposed algorithm which combined FCDD and FMD with ETP is examined. The FCDD and FMD with ETP combination reduce the total TS by 61.88% at the cost of 0.90% bit-rate increase, on average. BD-PSNR decrease also becomes tiny at -0.021dB. Our proposed algorithm shows very high performance.

In Table 3.9, the performance of our proposed algorithm is compared with the proposed algorithm in [28]. The previous work is the best performance and the state-of-the-art methodology of spatial scalability in SHVC. The test condition of the proposed algorithm is the same as the compared algorithm. The results given in Table 3.9 indicate that the proposed algorithm reduces TS 11.84% better than in [28] whereas it achieves better BD-BR as well. Hamid's algorithm [28] realize a good TS by bayes-decision rule. However, this approach requires high complexity calculation for RDO process. We consider that the hardware architecture including bayes-decision rule involve the increase of hardware are. In contrast, compared with [28] using bayes-decision rule, our proposed algorithm achieves the low computational complexity by simple approach. Therefore, our FCDD hardware can be implemented by very low cost.

Table 3.10 shows the simulation result of our proposed algorithm with common SHM test conditions. The QPs of the BL and EL were set to (QPBL, QPEL) = {(22,22), (26,26), (30,30), (34,34)} [33]. The verification method of [33] is newest for evaluation of SHVC. It is worth noting that the SHVC encoder selects the best chroma prediction mode among five modes including Planar, DC, Horizontal, Vertical and a direct copy of the IPM from the luma component. As a result, luma prediction performance affects U and V prediction performance as well. The results given in Table 3.10 indicate BD-rate comparison for coding of three color components by our proposed algorithm and the unmodified SHM11.0. From this Table 3.10, even if the high-resolution such as Class A and B is encoded in our proposed algorithm, the decreases of Y, U, and V do not induce at cost of 1.0% BD-rate with piecewise cubic. According to the above results, we confirm that the computational complexity reduction is achieved with almost no video quality loss.

Table 3.8 Result of proposed algorithm compared to SHM11.0

Class	Sequences	FCDD			FMD with ETP			FCDD and FMD with ETP		
		TS(%)	BD-BR(%)	BD-PSNR(dB)	TS(%)	BD-BR(%)	BD-PSNR(dB)	TS(%)	BD-BR(%)	BD-PSNR(dB)
Class 4K	Beauty	28.45	1.32	-0.001	40.56	2.28	-0.001	54.19	2.82	-0.001
	Bosphorus	30.25	0.35	-0.012	42.15	0.44	-0.015	56.33	0.65	-0.019
	HoneyBee	29.55	0.11	-0.001	42.03	0.12	-0.001	57.45	0.20	-0.002
	Jockey	28.36	0.75	-0.013	41.32	0.88	-0.016	55.80	1.16	-0.019
	ReadySetGo	27.95	0.66	-0.012	40.35	0.90	-0.014	55.41	1.06	-0.017
	ShakeNDry	30.05	0.02	-0.001	42.12	0.04	-0.001	58.26	0.05	-0.001
	YachtRide	28.96	0.96	-0.023	40.58	1.33	-0.047	57.61	1.86	-0.066
Class A	PeopleOnStreet	35.74	0.75	-0.030	43.74	0.55	-0.050	62.96	0.93	-0.065
	Traffic	37.15	0.68	-0.019	40.23	0.35	-0.020	63.01	0.71	-0.037
Class B	BasketballDrive	38.39	0.54	-0.010	46.25	0.69	-0.011	61.00	0.73	-0.014
	BQTerrace	39.44	0.57	-0.021	40.30	0.78	-0.036	59.61	0.92	-0.052
	Cactus	39.67	0.22	-0.009	44.21	0.34	-0.011	60.36	0.66	-0.026
	Kimono	40.05	0.18	-0.006	46.41	0.23	-0.012	63.36	0.36	-0.014
	ParkScene	40.76	0.19	-0.005	45.55	0.22	-0.013	64.66	0.43	-0.018
Class C	BasketballDrill	45.15	0.40	-0.023	55.34	0.50	-0.047	66.45	0.59	-0.053
	BasketballDrillText	42.12	0.48	-0.043	52.74	0.44	-0.055	63.50	0.64	-0.063
	PartyScene	40.20	0.44	-0.077	50.03	0.25	-0.068	61.30	0.52	-0.083
	RaceHorses	43.53	0.38	-0.049	51.29	0.33	-0.044	64.82	0.54	-0.056
Class D	BasketballPass	40.11	0.32	-0.031	48.35	0.44	-0.062	66.17	0.51	-0.070
	BQSquare	43.26	0.22	-0.055	49.24	0.39	-0.080	62.76	0.47	-0.097
	BlowingBubbles	41.99	0.22	-0.021	47.94	0.17	-0.050	59.57	0.25	-0.047
Class E	FourPeople	40.90	0.13	-0.059	45.86	0.10	-0.070	67.17	0.15	-0.082
	Johnny	43.21	0.56	-0.044	43.77	0.44	-0.045	61.81	0.60	-0.066
	KristenAndSara	45.11	0.39	-0.055	46.24	0.31	-0.033	71.79	0.52	-0.057
	SlideEditing	40.89	0.43	-0.032	42.74	0.59	-0.040	62.43	0.63	-0.043
	SlideShow	40.46	0.52	-0.033	43.33	0.47	-0.021	65.42	0.60	-0.044
	Vidyo1	45.92	0.54	-0.069	47.20	0.48	-0.079	69.20	0.66	-0.080
	Average	39.60	0.50	-0.051	45.18	0.68	-0.032	61.88	0.90	-0.021

Table 3.9 Comparison with other paper in time saving, BD-BR and BD-PSNR

	Hamid's algorithm[28]			Proposed algorithm (FCDD and FMD with ETP)			[28] vs Proposed algorithm		
Sequences	TS	BD-BR	BD-PSNR	TS	BD-BR	BD-PSNR	ΔTS	ΔBD-BR	ΔBD-PSNR
	(%)	(%)	(dB)	(%)	(%)	(dB)	(%)	(%)	(dB)
PeopleOnStreet	50.22	0.23	-0.013	62.96	0.93	-0.065	12.74	0.70	-0.052
Traffic	62.93	0.20	-0.010	63.01	0.71	-0.037	0.08	0.51	-0.027
BasketballDrive	50.82	1.87	-0.055	61.00	0.73	-0.014	10.18	-1.14	0.041
BQTerrace	50.82	0.84	-0.034	59.61	0.92	-0.052	8.79	0.08	-0.018
Cactus	58.27	0.57	-0.019	60.36	0.66	-0.026	2.09	0.09	-0.007
Kimono	58.71	0.19	-0.007	63.36	0.26	-0.014	4.65	0.07	-0.007
ParkScene	53.39	0.41	-0.017	64.66	0.43	-0.018	11.27	0.02	-0.001
BasketballDrill	43.91	1.52	-0.076	66.45	0.59	-0.053	22.54	-0.93	0.023
BlowingBubbles	45.08	1.13	-0.053	59.57	0.25	-0.047	14.49	-0.88	0.006
BasketballPass	41.82	1.55	-0.088	66.17	0.51	-0.070	24.35	-1.04	0.018
BQSquare	43.67	0.17	-0.013	62.76	0.47	-0.097	19.09	0.30	-0.084
Average	50.87	0.79	-0.035	62.71	0.58	-0.044	11.84	-0.20	-0.009

Table 3.10 Result of proposed algorithm (FCDD and FMD with ETP) compared to SHM11.0

Class	Sequences	Scalable layer	TS(%)	BD-rate (piecewise cubic)		
				Y(%)	U(%)	V(%)
Class A	PeopleOnStreet	EL	64.56	0.7	0.5	0.7
		Total	62.96	0.9	0.8	0.9
	Traffic	EL	65.17	1.2	1.4	1.1
		Total	63.01	1.1	1.2	1.0
Class B	BasketballDrive	EL	63.23	1.9	1.7	1.8
		Total	61.00	2.4	2.0	2.8
	BQTerrace	EL	62.41	1.3	1.2	1.5
		Total	59.61	1.3	1.2	1.4
	Cactus	EL	64.20	1.3	1.4	1.7
		Total	60.36	1.3	1.4	1.7
	Kimono	EL	67.25	0.4	0.7	0.7
		Total	63.36	0.6	0.8	0.8
	ParkScene	EL	67.37	0.4	0.7	0.7
		Total	64.66	0.3	0.6	0.6
	Average	EL	64.88	0.9	0.9	1.0
		Total	62.13	0.9	1.0	1.1

3.1.5 Hardware implementation

In this section, the proposed hardware implementation is represented. Firstly, over-all pipelined implementation of HEVC and the proposed algorithm are discussed. Additionally, the proposed algorithm shows improvements of hardware performance. Secondly, in the proposed FCDD hardware, the proposed an efficient address generator and a boundary calculation module are represented. Finally, synthesis results are shown and compared with another previous work.

3.1.5.1 Hardware implementation scheme

In previous works, several CTU-level paralleled hardware architectures for HEVC is achieved [23][24][25]. From these implementation results, it is considered reasonable that the overall pipelined implementation of HEVC is performed with about 300MHz working clock frequency. Based on these previous works, a top-level block diagram of HEVC is shown in figure 3.7 and a typical pipeline processing in HEVC is shown in figure 3.8. All-intra HEVC modules consist of intra prediction (IP), DCT, quantizer (Q), RDO, dequantizer (DQ), inverse DCT (IDCT), deblocking filter (DF), and adaptive loop filter (ALF).

To achieve the real-time processing in the typical case of 300MHz working clock frequency, the required number of cycles is given by

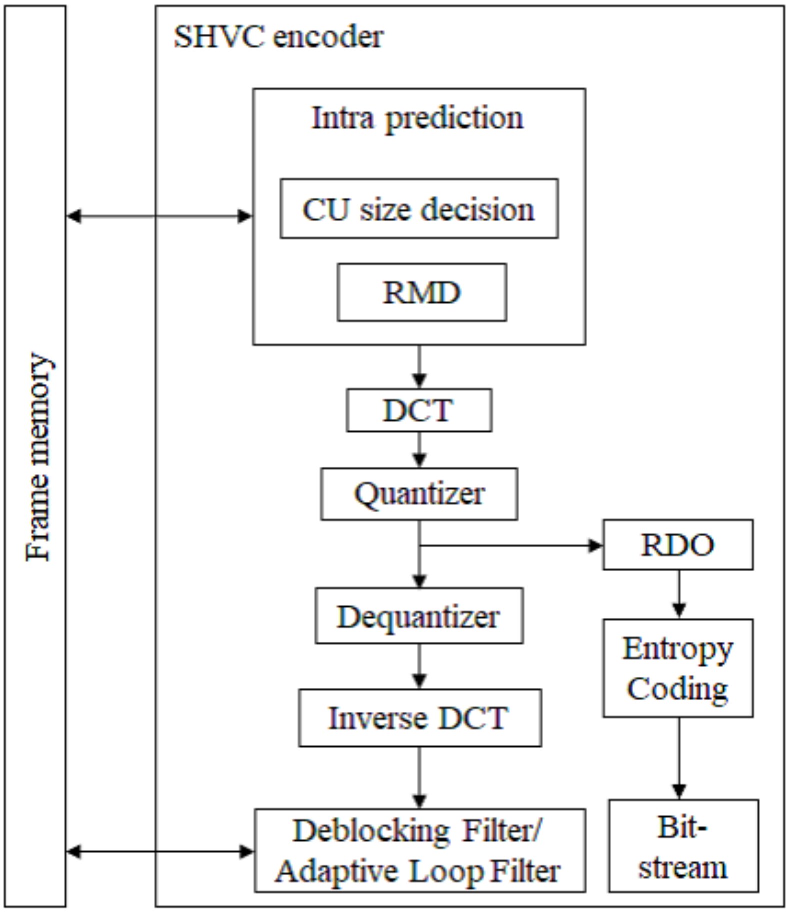


Figure 3.7 Top-level block diagram of HEVC.

$$\begin{aligned} Cycle_{fr} &= \frac{1}{60} \times 300 \times 10^6 \\ &= 5 \times 10^6 (Cycles/frame) \end{aligned} \tag{3.17}$$

$$\begin{aligned} Cycle_{CTU} &= \frac{Cycle_{fr}}{Num_{CTU}} \\ &= 2469 (Cycles/CTU) \end{aligned} \tag{3.18}$$

where $Cycle_{fr}$ represents the number of the total cycles in a frame and $Cycle_{CTU}$ represents the number of the total cycles in a CTU. The Num_{CTU} represents the number of the total CTUs and it is $(3,840x2,160)/(64x64)$. All of module need to limit the number of cycles to 2469 or less. Generally, the most complexity intensive processing is intra prediction (IP) in HEVC. Therefore, we focus on the discussion of the hardware implementation of RMD and RDO in our proposed algorithm. The

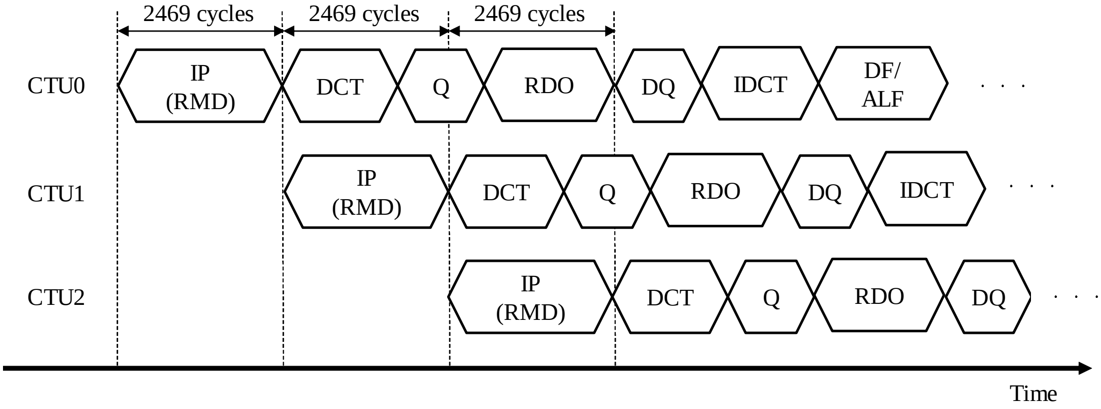


Figure 3.8 CTU-level pipeline scheduling for all-intra HEVC.

benefits of the proposed algorithm for hardware implementation can be concluded as high parallel-processing, small area design, and low working clock frequency.

As a pre-processing algorithm, the proposed FCDD can completely be pipeline processed with the traditional encoding process. Furthermore, a CTU-level pipeline architecture can be applied to RMD and RDO mode decision as shown in figure 3.9. Accordingly, the RMD and RDO stage can be operated in parallel. With a highly paralleled architecture, an exhausted RDO processing load is separated to FCDD and RMD. As the results, many clock cycles can be reduced for the intra coding in SHVC.

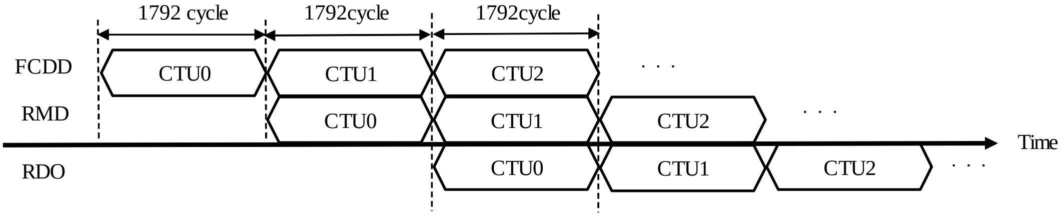


Figure 3.9 CTU-level pipeline scheduling for FCDD, RMD and RDO.

Regarding the area cost of the proposed FCDD, some additional hardware cost is necessary for the calculation of boundary correlation. However, since the boundary calculation of the vertical line and the horizontal line is very simple, only very small area hardware overhead is required. Moreover, comparing with another previous work which also introduced a pre-processing Sobel filter to detect the edge, the proposed algorithm use only the pixels on the edge of CUs instead of all the pixels in a CTU [20]. Therefore, it is clear that the proposed algorithm can easily realize hardware area efficiency than the previous work.

Regarding the hardware architecture of intra prediction, when implementing FCDD and FMD with ETP, the RMD and RDO process in a worst case is obtained as 1,792 cycles, which is calculated by

$$\begin{aligned} Cycle_{CTU} &= \frac{64}{4} \times \frac{64}{4} \times 7 \\ &= 1,792(Cycles/CTU) \end{aligned} \tag{3.19}$$

in which it assumes that 1 mode can be performed in 1 cycle. The worst case is considered as when the number of IPM is 7 and all CUs are encoded by 4x4. The calculation result shows that no more than 1,792 cycles are permitted to encode a CTU. On the basis of this calculation, the minimum operation clock frequency (F_{min}) of RMD mode decision for real-time processing is given by

$$\begin{aligned} F_{min} &= Cycle_{CTU} \times Num_{CTU} \times fr \\ &= 219.34(MHz) \end{aligned} \tag{3.20}$$

where Num_{CTU} represents the number of the total CTUs and fr is the encoding frame rate in one frame, respectively. In the implementation of our algorithm, $Cycle_{CTU}$ is 1,792. For a 4k@60fps real-time processing hardware, the Num_{CTU} is $(3,840\text{x}2,160)/(64\text{x}64)$ and the typical value of fr is 60. Substituting the corresponding values, as shown by equation (3.20), only 220 MHz is required to achieve real-time processing.

3.1.5.2 Overview of FCDD hardware architecture

The previous section shows that the number of required cycles to achieve a CTU-level pipeline architecture is less than 1,792. In other words, the proposed FCDD processing needs to be completed within 1792 cycles. Now, the number of cycles for vertical line in the proposed FCDD algorithm is estimated by

$$Cycle_{BC_V} = Cycle_{BC_H}$$

$$\begin{aligned} &= 8 \times \frac{64}{8} \times \frac{64}{8} + 16 \times \frac{64}{16} \times \frac{64}{16} \\ &\quad +32 \times \frac{64}{32} \times \frac{64}{32} + 64 \times \frac{64}{64} \times \frac{64}{64} \\ &= 960(Cycles/CTU) \end{aligned} \tag{3.21}$$

where the calculation using neighboring two pixels of the boundary can be performed in 1 cycle, and the multiplication is represented by the number of the boundary calculation from 8x8 to 64x64 in a CTU. The number of cycles of horizontal line in the FCDD algorithm can be also obtained as well as that of the vertical line ($Cycle_{BC_H}$). Considering a CTU-level pipeline architecture, if the boundary calculation of the vertical line and the horizontal line are performed with the serial-processing, the boundary calculation requires 1,920 cycles. Accordingly, the boundary calculation with the serial-processing finds it hard to achieve real-time processing of 4k@60fps hardware from the previous section. This problem can be solved by designing the parallel-processing of the vertical line and the horizontal line. However, there is the possibility that the parallel-processing involves the increase of hardware cost. Therefore, an efficient FCDD hardware architecture is proposed.

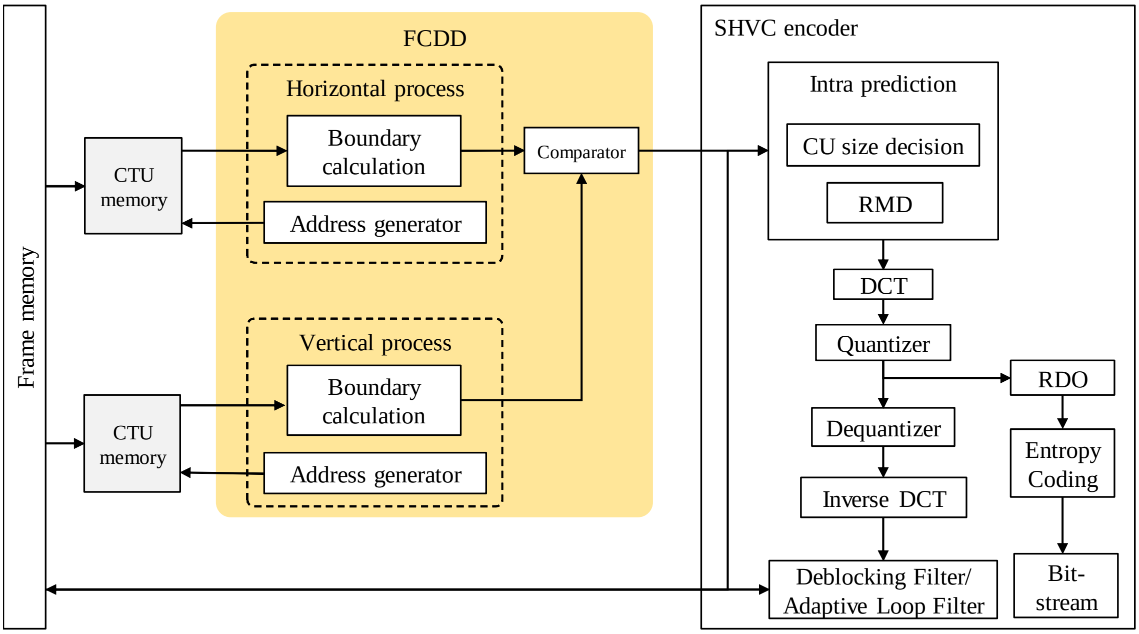


Figure 3.10 Overview of the FCDD hardware architecture processing.

Overview of the FCDD hardware architecture is shown in figure 3.10. In our

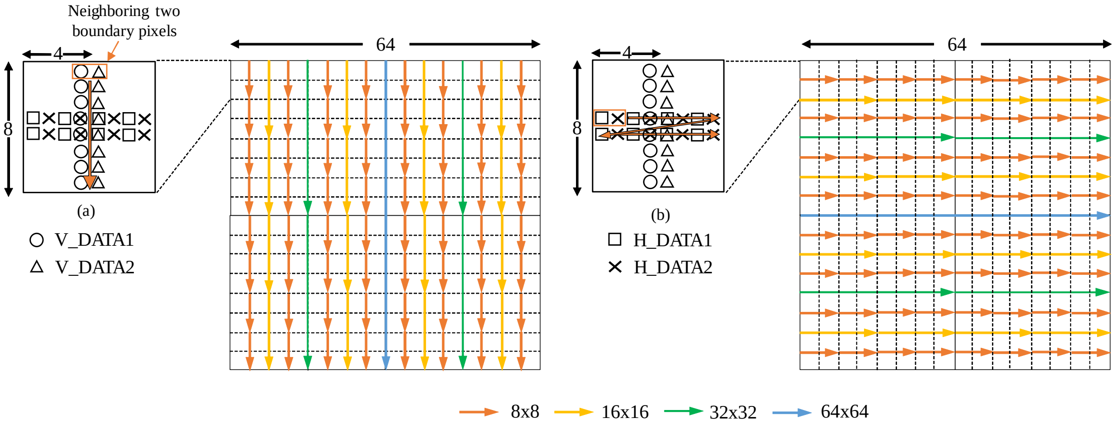


Figure 3.11 Processing order of vertical line (a) and horizontal line (b).

scheme, the pixel data is stored from the frame memory to the CTU memory. The number of cycles to input from the frame memory to the CTU memory and SHVC encoder is sufficiently ensured by using the parallel boundary calculation. Moreover, to read the pixel data which is used for the boundary calculation from the CTU memory, an address generator is designed. If the parallel-processing of the vertical line and the horizontal line is used for boundary calculation, the input data for several kinds of CU size need to be carefully arranged. The FCDD hardware architecture which is composed with the boundary calculation and the address generator can provide the complete pipeline processing with the intra coding in SHVC. The detail of the address generator and the parallel boundary calculation are introduced in the next section.

3.1.5.3 Efficient address generator

In our proposed hardware architecture, the address generator which can generate one word (2 bytes) for 1 address is designed. By using the generated address, the neighboring two pixel data are input from the CTU memory to the boundary calculation module. Accordingly, the proposed FCDD hardware architecture can access the neighboring two boundary pixels from 1 address as shown in figure 3.11.

To achieve efficient memory access, the transposition conversion is required for the CTU memory of the vertical line. By applying the transposition conversion

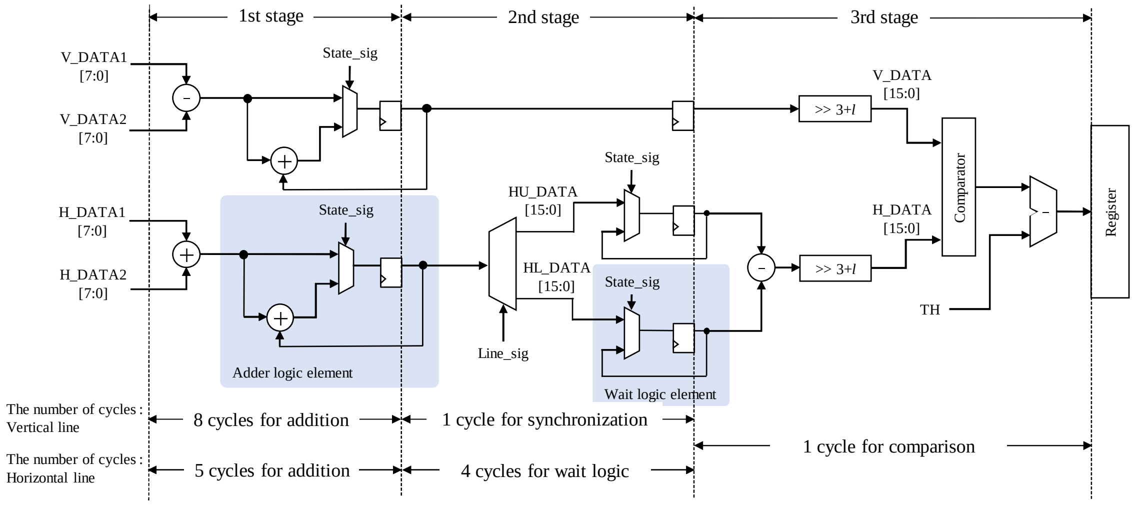


Figure 3.12 The proposed hardware architecture for the boundary calculation module in the case of block size with 8x8.

to the CTU memory, the required pixels for boundary calculation can be input by simple addressing. However, the transposition conversion consumes a large number of cycles, making the parallel-processing between the horizontal line and the vertical line difficult. Additionally, the hardware cost is increased by the implementation of the transposition conversion. Therefore, we consider that the hardware implementation of transposition conversion is redundant for FCDD hardware. For these reasons, our proposed address generator achieve the parallel-processing without transposition. As shown in figure 3.11, the boundary calculation use the neighboring two boundary pixels. The method of using the neighboring two boundary pixels in the CTU memory allow a simple addressing. The simple addressing provides the complexity reduction of the address generator. Based on the above, from the view point of the parallel-processing and hardware cost, the most efficient address generator is designed. Moreover, the most suitable boundary calculation module for the proposed address generator is represented by next section.

3.1.5.4 The feature of the boundary calculation module

The processing order in the boundary calculation module is shown by figure 3.11. The processing order of the vertical line (a) is performed from upside to downside using neighboring two pixels of the boundary. On the other hand, the processing order of the horizontal line (b) is performed from left side to right side using neighboring two pixels, and this processing includes return. As shown in figure 3.11, V_DATA1, V_DATA2, H_DATA1, and H_DATA2 are simultaneously input for the boundary calculation module in every 1 cycle.

The detail of the proposed hardware architecture for the boundary calculation is shown in figure 3.12. The hardware architecture is composed with three stage. The proposed boundary calculation module is implemented with highly parallel-processing and few number of cycles. In the first stage, the additional and subtraction process are performed with each line. Additionally, the adder logic elements are used for boundary calculation of each line. By using state machines, the boundary calculation in the block size from 8x8 to 64x64 is controlled. The signal from state machines (State_sig) controls the selector logic. In the case of 8x8 block, V_DATA and H_DATA consume 8 cycles and 5 cycles, respectively. In the second stage, V_DATA consumes 1 cycle to synchronize with H_DATA. On the other hand, because the boundary calculation of H_DATA includes return processing, the upper line (HU_DATA) and the lower line (HL_DATA) have to be controlled. Accordingly, HU_DATA and HL_DATA are switched by demultiplexes. Moreover, the wait logic elements store the value until setting HU_DATA and HL_DATA. The wait logic elements consume 4 cycles. In the third stage, the average value is calculated with each V_DATA and H_DATA, and a comparator compares V_DATA and H_DATA. The comparator output indicates the maximum value of boundary calculation. The maximum value is compared with TH value, as shown in Table 3.1. Then, the information of the optimal CU size in a CTU is stored to register. The comparison processing consumes 1 cycle.

Figure 3.12 represents the number of cycles in the boundary calculation in the case of 8x8 block. If the boundary calculation is performed with the other block size, the number of cycles in adder logic element and wait logic is controlled by State_sig. Table 3.11 shows the number of required cycles in each stage when performing the other block size. Then, N represents the block size in which the boundary calculation is performed. As shown in Table 3.11, even though the boundary calculation is performed with each block size, the boundary calculation module achieve highly parallel-processing.

Table 3.11 The number of required cycles in each stage

	1st stage	2nd stage	3rd stage
Vertical line	N	1	1
Horizontal line	$N/2+1$	$N/2$	1

Regarding the hardware cost, the boundary calculation module uses two pixels as input signals in our proposed hardware architecture. If the input signal is added to reduce the number of cycles in the boundary calculation module, the hardware cost is increased because additional logic cells are required by addition and subtraction. In this case, there are logic cells which are not used for the boundary calculation depending on the CU size. For example, when the boundary calculation module uses sixteen pixels as input signals, redundant logic cells arise with the boundary calculation of 8x8. Therefore, by using two pixels as input signals, the boundary calculation module achieve low implementation cost. Moreover, considering the compatibility between the address generator and the boundary calculation module, the number of cycles is not consumed wastefully. In other words, FCDD hardware architecture achieves the efficient memory access.

3.1.5.5 Implementation result

The proposed FCDD hardware is described by using Verilog HDL and synthesized on the FPGA platform, and synthesis results are given in Table 3.12. From synthesis result in Table 3.12, the hardware achieves scalable working clock frequency with less than 6% of the total resources consumed.

As one of the purposes of this work, the proposed FCDD hardware architecture is mainly used to determine the optimal CU depth decision before encoding. In previous work, the hardware cost of HEVC encoder shows 1571.7K gate [24]. Comparing the HEVC hardware encoder, the proposed FCDD hardware is implemented with very low cost. Therefore, the proposed FCDD hardware architecture indicate the beneficial pre-processing for HEVC encoder. Moreover, we confirmed that the proposed FCDD hardware can be adapted to various working clock frequency from Table 3.12. By achieving high parallel-processing, the proposed FCDD hardware can be performed with 120MHz working clock frequency. For example, when achieving HEVC encoder design that aims to low power consumption, a CTU need to be encoded by the low working clock frequency. In this case, because the recursive RDO process in HEVC encoder can be reduced by adding our proposed hardware to HEVC encoder, the proposed FCDD hardware is very useful for HEVC encoder design with low working clock frequency. Additionally, when high-resolution video (8k@60fps) is encoded, the high working clock frequency is required for HEVC hardware encoder. The proposed FCDD hardware can be also performed with the high working clock frequency at 416MHz. Therefore, the proposed FCDD hardware can be flexibly applied to various HEVC hardware encoder.

The proposed FCDD hardware is also compared with another previous work [22], as shown in Table 3.12. As for the total logic utilization, in the previous work, the total logic utilization represents the number of logic which is used to only the edge detection. Therefore, considering the core hardware architecture which is only the edge detection, the number of logic for the proposed FCDD hardware is implemented by a few logics at 279 gates. Regard to the comparison of the critical path, max frequency of the proposed hardware achieve high working clock frequency compared to previous work because logic counts are small. In contrast, a large logic counts of previous work have a long critical path because the hardware architecture requires the computational complexity using all input pixels for the edge detection. The number of input pixels in the proposed FCDD architecture is 1920 in the case of parallel processing in vertical line and horizontal line Therefore, the proposed FCDD architecture reduce 53.12% input pixels in a CTU, which is calculated by $(4096 - 1920)/4096$. Moreover, the coding performance is improved with 0.5% BD-BR compared with [22].

Table 3.12 Synthesis result of the proposed pre-encoding hardware architecture and comparison of the previous work

Our proposed hardware architecture		Previous work[22]	
FPGA (Cyclone V)	5CGXFC5C6F27C7N	CMOS technology	0.18 μm standard cell library
Frequency (MHz)	416MHz	Max frequency	235 MHz
-Total Logic utilization	1,807	Total Logic utilization	1,659
-Address generator	1,528/29,080 (<1%)	-	-
-Combinational Functions	1,432/29,080 (<6%)	-	-
-Total registers	96/29,080 (<1%)	-	-
-Boundary calculation	279/29,080 (<1%)	-	-
-Combinational Functions	151/29,080 (<6%)	-	-
-Total registers	128/29,080 (<1%)	-	-
Frequency (MHz)	120MHz	-	-
-Total Logic utilization	1,809	-	-
-Address generator	1,530/29,080 (<1%)	-	-
-Combinational Functions	1,434/29,080 (<6%)	-	-
-Total registers	96/29,080 (<1%)	-	-
-Boundary calculation	279/29,080 (<1%)	-	-
-Combinational Functions	151/29,080 (<6%)	-	-
-Total registers	128/29,080 (<1%)	-	-
Profile	Main, Scalable extension	Profile	Main
Resolution	4K@60fps	Resolution	8K@60fps
BD-BR (%)	0.9	BD-BR (%)	1.4
TS (%)	38.7	TS (%)	45.8

3.1.6 Conclusion

The focus of this paper is on developing a complexity reduction scheme for spatial scalable SHVC encoder. The proposed algorithm uses fast CU depth decision (FCDD), fast mode decision (FMD), and early termination process (ETP). The performance of the proposed algorithm was tested over a representative set of video sequences and was compared to the unmodified SHVC encoder as well as two of the art complexity reduction schemes and combinations. Performance evaluations show that our proposed algorithms reduce encoding time on average 61.88% and increases BD-rate about 0.9%, compared with SHM 11.0. Moreover, to confirm a validity of the proposed FCDD algorithm, the hardware architecture is designed targeting on the FCDD algorithm. Synthesis results show that the hardware cost is about 1.8K gates and achieve the scalable working clock frequency in the case of FPGA (CycloneV) implementation.

3.2 3D extension of HEVC

3.2.1 Research motivation in 3D extension

With the development of the technology of 3D television (3DTV) and free viewpoint television (FTV), 3D video coding (3D-HEVC) attracts more attention. The typical 3D video is represented using the multi-view video and depth format [38], in which few captured texture videos as well as associated depth maps are used. The depth maps provide per-pixel with the depth corresponding to the texture video that can be used to render arbitrary virtual views by using depth image based rendering [39][40]. Recently, 3D-HEVC technology based on high efficiency video coding (HEVC) is now being standardized by joint collaborative team on 3D video coding (JCT-3V) as an extension to HEVC [41][42][43]. From the JCT-3V meetings, the developed coding schemes for 3D-HEVC mainly use HEVC together with exploiting temporal and interview correlation. Thus, many coding tools applied in 3D-HEVC are based on the hybrid coding scheme and highly related to HEVC. Considering the intra depth coding, the traditional HEVC prediction modes result in distinct coding artifacts at sharp edges. To represent the depth information in a better way, two depth modeling modes (DMMs), named wedgelet partition mode (DMM1) and contour partition mode (DMM4), have been added to the intra coding of 3D-HEVC. DMMs can largely contribute to the depth coding, but much computational complexity is induced. Moreover, the decisions of the coding units (CUs) and modes have the most of the computational complexity for HEVC encoder. The 3D-HEVC also adopts the quad-tree structure coding, which supports the coding units (CUs) varying from 64x 64 to 8x8 (to 4x4, if considering PU partition). The traditional process of CU decision includes very high computational complexity. For the mode decision, it takes more computation time due to DMMs. The complexity of intra depth coding takes about 5 times more than that of texture coding and contributes to the coding efficiency for 3D-HEVC [44]. [45] also shows the detail of the complexity distribution according to the CU size, and it is clear that two DMMs occupies a large proportion among the complexity distribution of encoding. Therefore, the low complexity algorithms for intra depth coding are required.

Some previous works have proposed the complexity reduction for the intra depth coding in 3D-HEVC [46]-[54]. In [46] and [47], Gu et al. used the evaluation cost of intra traditional mode as the skip signal to avoid some DMMs calculation. Park et al. proposed an algorithm which performs a simple edge classification in the Hadamard transform to omit unnecessary DMMs [48]. In [49], Silva et al. proposed an algorithm to reduce the number of modes to be evaluated in the mode decision. All these methods focus on the mode decision of DMMs. Thereby, improvement of DMMs is discussed positively. Regarding fast CU decision algorithm, [50] and [51] proposed the selection of the adaptively CU and intra prediction mode from rate-distortion (RD) cost. However, further improvement is required for the low complexity algorithm because the complexity reduction with the high bit-rate tend to be low complexity reduction rate. Moreover, Sanchez et al. proposed an aggressive and lightweight complexity reduction technique using the simplified edge detector (SED) algorithm [52][53]. The one presented in [52] is capable of performing bipartition modes evaluation without data dependencies in HEVC intra-prediction mode, which is a desired characteristic in a hardware design. Additionally, [53] presents the development of the SED hardware design. This related work contributes to a hardware oriented algorithm. However, SED algorithm has to perform the filter process for all pixel information in a CTU. Considering hardware implementation of pre-processing, the hardware architecture area has been increased in the method of [53]. Therefore, the efficient edge detection which can determine the optimal CU depth is required for hardware implementation of pre-encoding. As a particularly new technique, H. Zheng et al. proposed [54] a low complexity depth intra coding method for 3D-HEVC based on the depth classification. For optimal CU decision, the classifier trained by support vector machine (SVM) is applied to determine its depth complexity class for only checking the corresponding intra prediction modes. [54] is also very effective for the reduction of recursive rate-distortion optimization (RDO) process. However, [54] consume the high computational time by performing the depth classification. Therefore, it is difficult to apply the depth classification using SVM to real-time encoding processing.

Our research target is on the complexity reduction for RDO process in 3D-

HEVC. Additionally, to achieve the hardware oriented parallel algorithm without data dependencies in RDO process, high efficient pre-processing is proposed.

3.2.2 Overveiw of 3D HEVC

The 3D-HEVC standard inherits the well-known block based hybrid coding architecture of HEVC. It employs a flexible quad-tree coding block partitioning structure that enables the usage of large and multiple sizes of CU. One of the frames is divided into a sequence of coding tree units (CTUs) and the maximum size allowed for the luma block in a CTU is specified to be 64x64. Each 2Nx2N CUs which shares the same prediction mode can be divided into four smaller NxN CUs recursively until the minimum CU size is reached. The sizes of CU range from 64x64 to 8x8. The number of the prediction modes in intra depth coding for each CU is also increased to 37, including 35 conventional modes in HEVC, and two DMMs. In the DMMs, two different types of partitioning patterns named wedgelets and contours are applied to satisfy the characteristics of depth maps. They are different in the segmentation of the depth block. Each pattern of partition divides the area of the block into two nonrectangular regions, where each region is represented by a constant value.

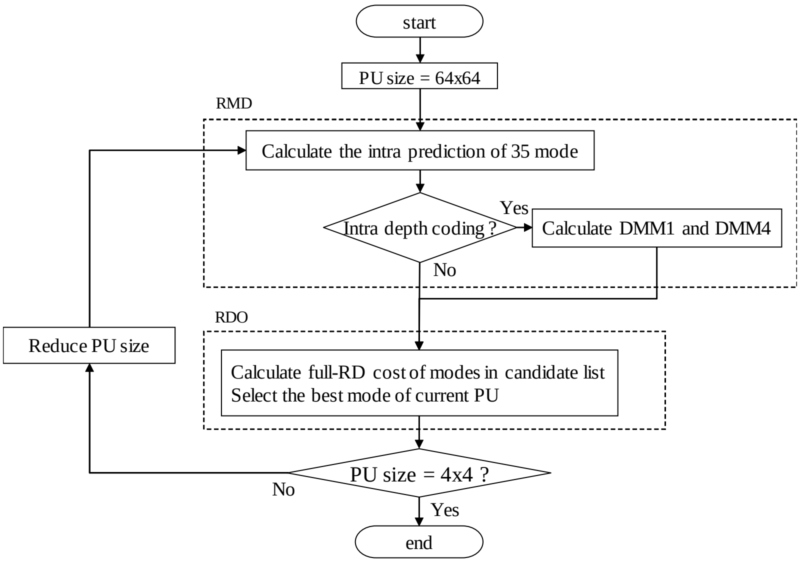

start
PU size = 64x64
RMD
Calculate the intra prediction of 35 mode
Intra depth coding ?
Yes
Calculate DMM1 and DMM4
No
RDO
Reduce PU size
Calculate full-RD cost of modes in candidate list
Select the best mode of current PU
PU size = 4x4 ?
No
Yes
end

Figure 3.13 The processing flow of intra coding in HTM16.0.

The flowchart of intra texture and depth coding in the reference software 3D-HEVC Test Model (HTM16.0) is introduced [41][42], as shown in figure 3.13. For each iteration, it starts from trying the CUs size of 64x64. The CU size ranges from 64x64 to 8x8. For each CU, the mode decision could be divided into rough mode decision (RMD) and rate-distortion optimization (RDO) stages. In the RMD process, 35 HEVC intra prediction modes are searched with the sum of absolute transformed difference. Three or eight modes are selected as the candidate modes. If intra depth coding is performed, DMMs with all partitions are searched with the sum of squared error. The best DMMs are put into candidate modes list as well. In the RDO process, the candidate modes are searched with the complex full RD cost function, where the evaluation cost is defined as JRDO. JRDO highly depends on the quantization parameter (QP). The quantizer of HEVC is similar to that of H.264/AVC where QP is defined in the range of [0, 51]. An optimized bit-rate can be generated by adaptively selecting the QP. The best mode is decided with the smallest JRDO and this cost is regarded as the current CU cost. For the CU decision, the cost of each CU is always compared with the total cost of its four sub-CUs to decide whether the CU is to be split or not.

3.2.3 Proposed algorithm

3.2.3.1 Efficient edge detection by Laplacian filter and edge classification for intra depth prediction mode

The edge detection in [48] and [52] need to perform Hadamard transform and the simplified edge for all pixel information in a CTU. On other words, these edge detection processes induce the recursive computational complexity. To achieve more computation complexity reduction than other edge detection algorithm, we propose an efficient edge detection algorithm for fast intra depth prediction mode. In our approach, the adaptive Laplacian filter (LF) processing is performed while calculating BHs of ECDD. In the previous section, the strength of the boundary homogeneity of the horizontal line and the vertical line are used as $BH(k,m,l)$. Additionally, by calculating $DiffBH(k,m,l,i)$, the boundary position of depth image can be obtained. The detail of the boundary position is calculated by following equation.

$$DiffBH_1(k,m,l,i) = \\ |P(x_{2^{2+l}-1+2^{3+l}(k-1)}, y_{i+2^{3+l}(m-1)}) - P(x_{2^{2+l}-1+2^{3+l}(k-1)}, y_{i+1+2^{3+l}(m-1)})| \quad (3.22)$$

$$DiffBH_2(k,m,l,i) =$$

$$|P(x_{2^{2+l}+2^{3+l}(k-1)}, y_{i+2^{3+l}(m-1)}) - P(x_{2^{2+l}+2^{3+l}(k-1)}, y_{i+1+2^{3+l}(m-1)})| \tag{3.23}$$

$$DiffBH_3(k,m,l,i) =$$

$$|P(x_{i+2^{3+l}(k-1)}, y_{2^{2+l}-1+2^{3+l}(m-1)}) - P(x_{i+1+2^{3+l}(k-1)}, y_{2^{2+l}-1+2^{3+l}(m-1)})| \tag{3.24}$$

$$BH_4(k,m,l,i) =$$

$$|P(x_{i+2^{3+l}(k-1)}, y_{2^{2+l}+2^{3+l}(m-1)}) - P(x_{i+1+2^{3+l}(k-1)}, y_{2^{2+l}+2^{3+l}(m-1)})| \tag{3.25}$$

$DiffBH_1$, $DiffBH_2$, $DiffBH_3$, and $DiffBH_4$ represent the difference value of the neighboring pixel. i represent the position of $DiffBH$. In our approach, while the $BH(k,m,l,i)$ are obtained, the calculation of $DiffBH$ are performed. Therefore, the proposed algorithm can detect the edge information efficiently. The detail of the edge detection method in the case of $k,m,l = 0$ is introduced by figure 3.14 and figure 3.15.

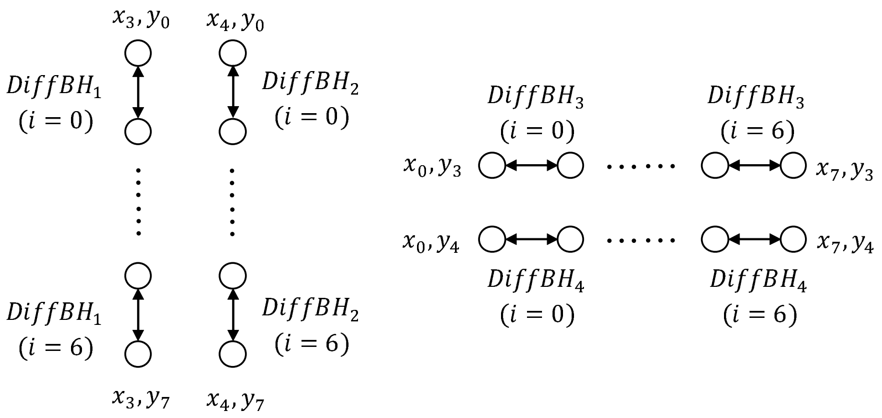


Figure 3.14 The representation of $DiffBH(k,m,l,i)$ from $i = 0$ to $i = 6$ in the case of $k = 0$, $m = 0$, and $l = 0$.

Figure 3.14 represents the position of $DiffBH(k,m,l,i)$. $DiffBH(k,m,l,i)$ is used to judge whether to apply LF. Therefore, our proposed edge detection algorithm achieves an adaptive LF selection. From some verification result, it is clear that the edge direction can be detected by LF when the difference value of the pixel is larger than 5. Accordingly, if the value of $DiffBH(k,m,l,i)$ is larger than 5, which is defined as edge point (EP), the coefficients are calculated around the EP by using the LF. For example, figure 3.15 shows the position where LF is applied when EP is $DiffBH_3(1,1,0,2)$. By using the calculated coefficients, the candidate mode list in $DiffBH_3(1,1,0,2)$ is selected from Table 3.13. Additionally, the candidate mode list in $DiffBH_3(1,1,0,2)$ is stored to $BH\_Edge(1,1,0,2)$. Similarly, $BH\_Edge(1,1,0,i)$ from i = 0 to i = 6 is calculated. Figure 3.16 shows the detail position of $BH_1\_Edge(k,m,l,i)$, $BH_2\_Edge(k,m,l,i)$, $BH_3\_Edge(k,m,l,i)$, and $BH_4\_Edge(k,m,l,i)$. As shown in figure 3.16, BH_Edge represent the candidate modes of boundary region. The candidate mode list of 4x4 block ($\mathbf{X}_{\text{4x4}}$) is represented by following.

$$\mathbf{X}^0_{\text{4x4}} = \begin{cases} BH_1\_Edge(k,m,l,i)\ (0 \leq i \leq 3) \\ BH_3\_Edge(k,m,l,i)\ (0 \leq i \leq 3) \end{cases} \tag{3.26}$$

$$\mathbf{X}^1_{\text{4x4}} = \begin{cases} BH_2\_Edge(k,m,l,i)\ (0 \leq i \leq 3) \\ BH_3\_Edge(k,m,l,i)\ (3 \leq i \leq 6) \end{cases} \tag{3.27}$$

$$\mathbf{X}^2_{\text{4x4}} = \begin{cases} BH_1\_Edge(k,m,l,i)\ (3 \leq i \leq 6) \\ BH_4\_Edge(k,m,l,i)\ (0 \leq i \leq 3) \end{cases} \tag{3.28}$$

$$\mathbf{X}^3_{\text{4x4}} = \begin{cases} BH_2\_Edge(k,m,l,i)\ (3 \leq i \leq 6) \\ BH_4\_Edge(k,m,l,i)\ (3 \leq i \leq 6) \end{cases} \tag{3.29}$$

In the case of 4x4 block, the edge classification values of eight which is selected from Table 3.13 are stored to $\mathbf{X}_{\text{4x4}}$. To decide the edge classification value used for the intra depth prediction mode, the most selected edge classification value in $\mathbf{X}^0_{\text{4x4}}$, $\mathbf{X}^1_{\text{4x4}}$, $\mathbf{X}^2_{\text{4x4}}$, and $\mathbf{X}^3_{\text{4x4}}$ are substituted for X^0_{4x4}, X^1_{4x4}, X^2_{4x4}, and X^3_{4x4}. Moreover, for calculating X_{8x8}, X_{16x16}, X_{32x32}, and X_{64x64}, the combinational method is introduced.

Table 3.13 Candidate list of the mode number

Edge classification	Mode number
DC, Planar	0, 1
Vertical	22, 23, 24, 25, 26, 27, 28, 29, 30
Horizontal	6, 7, 8, 9, 10, 11, 12, 13, 14
Left direction	2, 3, 4, 5, 30, 31, 32, 33, 34
Right direction	13, 14, 15, 16, 17, 18, 19, 20, 21

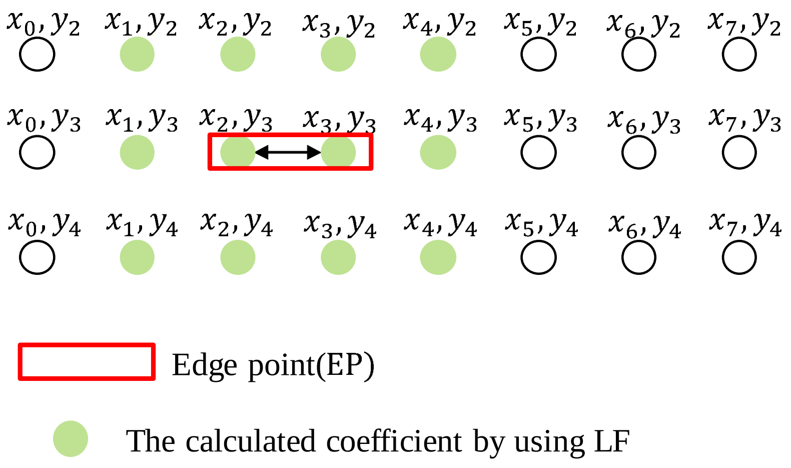


Figure 3.15 The position where LF is applied when EP is $DiffBH_3(1,1,0,2)$.

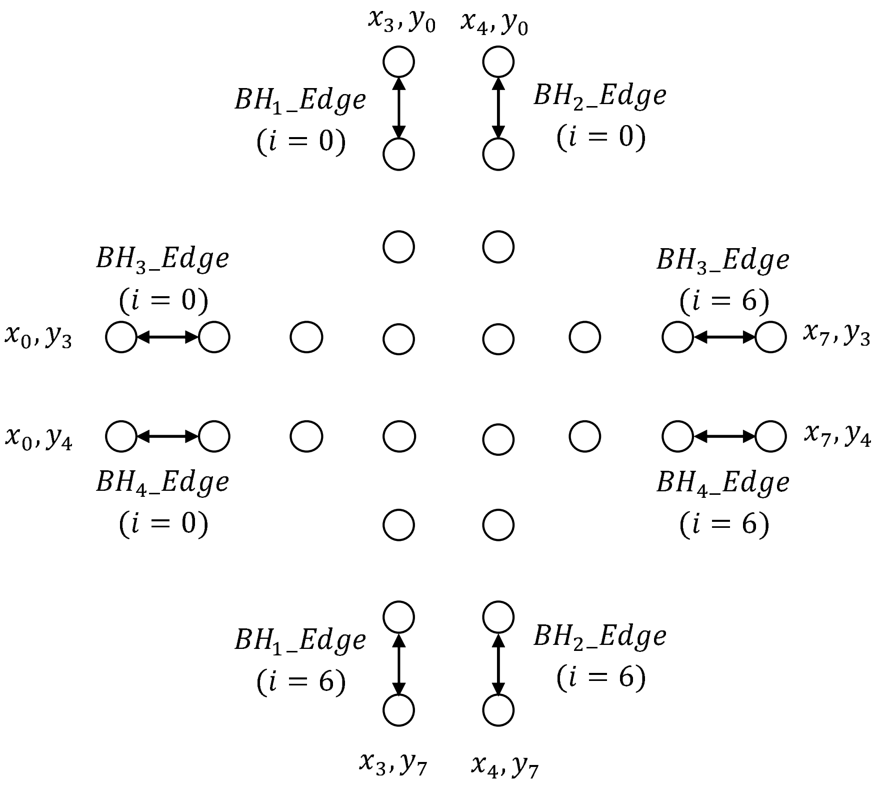


Figure 3.16 The representation of $BH\_Edge(k,m,l,i)$ in the case of $k=0$, $m=0$, and $l=0$

In the proposed algorithm, if more than two of the edge classification value in X^0_{4x4}, X^1_{4x4}, X^2_{4x4}, and X^3_{4x4} are same, the edge classification value is used as the prediction mode of $X^0_{8\text{x}8}$. Figure 3.17 represents the combinational case of $X^0_{2N\text{x}2\text{N}}$. Similarly, the edge classification value of 16x16, 32x32, and 64x64 block are calculated. The detail of the procedure is described in Algorithm 3 and Algorithm 4.

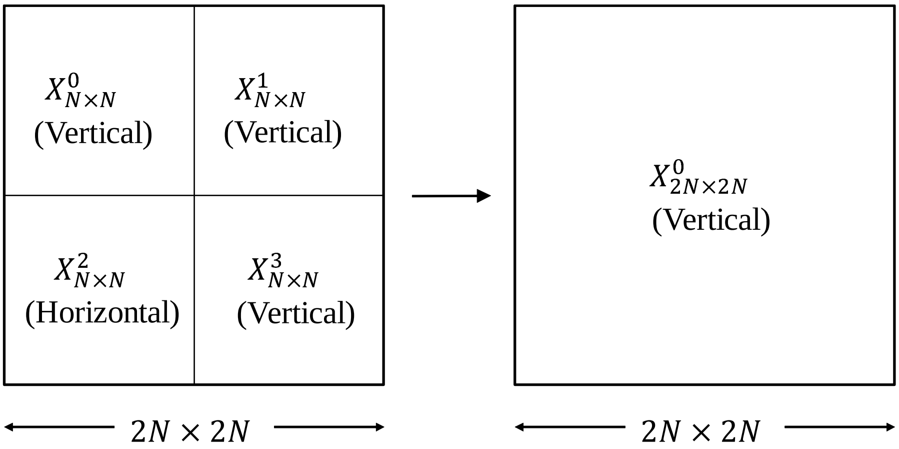


Figure 3.17 Graphical explanation of combinational case

Algorithm 3 show the efficient edge detection and the edge classification algorithm. The efficient edge detection is achieved by $Edge\_flag(i)$ (line 3-7). If $Edge\_flag(i)$ is equal to 1, Laplacian coefficients is calculated and the candidate list of the mode number is selected (line 8-10). On the other case, the candidate list of DC and Planar mode is selected (line 11-14). After calculating all of $BH\_Edge(k, m, l, i)$, the edge classification ($X_{4\text{x}4}$) is obtained (line 16-17).

As shown in Algorithm 4, The combinational algorithm is performed. $X^0_{2N\text{x}2\text{N}}$ is decided by $X^0_{N\text{xN}}$, $X^1_{N\text{xN}}$, $X^2_{N\text{xN}}$, $X^3_{N\text{xN}}$ and (line 1). The combinational case is explained by the previous paragraph (line 2-4). On the other case, $X^0_{2N\text{x}2\text{N}}$ is selected in order of vertical, horizontal, left diagonal, and right diagonal (line 5-15). This is because it is clear that this order is the most probably selected from previous work [51].

3.2.3.2 Overall processing

The proposed algorithm is composed of two parts intra texture and depth coding. Firstly, the boundary homogeneity with intra texture and depth coding respectively is calculated for optimal CU size decision. Compared with HTM16.0, the proposed

Algorithm 3 : Efficient edge detection and edge classification algorithm

```
if $l = 0$ then
    for $i = 0$ to $2^{3+l} - 2$ do
        if $DiffBH(k, m, l, i) > 5$ then
            $Edge\_flag(i) = 1$
        else
            $Edge\_flag(i) = 0$
        end if
        if $Edge\_flag(i) == 1$ then
            Laplacian coefficients are calculated as the center of $DiffBH(k, m, l, i)$.
            Using Laplacian coefficient, the candidate list of the mode number is selected, and stored to $BH\_Edge(k, m, l, i)$.
        else
            $BH\_Edge(k, m, l, i) = 0$
            DC and Planar mode are stored.
        end if
    end for
    Based on $BH\_Edge(k, m, l, i)$, the edge classification values of eight which is selected from Table 3.13 are stored to $\mathbf{X}_{4\text{x}4}$.
    $X_{4\text{x}4}$ is calculated from $\mathbf{X}_{4\text{x}4}$
end if
```

Algorithm 4 : Combinational algorithm

```
$X^0_{2N\text{x2N}} = \{X^0_{N\text{xN}}, X^1_{N\text{xN}}, X^2_{N\text{xN}}, X^3_{N\text{xN}}\}$ $(N = 4, 8, 16, 32)$
if Combinational case then
    More than two of the edge classification value in $X^0_{N\text{xN}}$, $X^1_{N\text{xN}}$, $X^2_{N\text{xN}}$, and $X^3_{N\text{xN}}$ are same.
    $X^0_{2N\text{x2N}} = X_{N\text{xN}}$
else
    if $X^0_{N\text{xN}}$, $X^1_{N\text{xN}}$, $X^2_{N\text{xN}}$, and $X^3_{N\text{xN}}$ include vertical. then
        $X^0_{2N\text{x2N}}$ = Vertical
    else if $X^0_{N\text{xN}}$, $X^1_{N\text{xN}}$, $X^2_{N\text{xN}}$, and $X^3_{N\text{xN}}$ include horizontal. then
        $X^0_{2N\text{x2N}}$ = Horizontal
    else if $X^0_{N\text{xN}}$, $X^1_{N\text{xN}}$, $X^2_{N\text{xN}}$, and $X^3_{N\text{xN}}$ include left diagonal. then
        $X^0_{2N\text{x2N}}$ = Left diagonal
    else$X^0_{N\text{xN}}$, $X^1_{N\text{xN}}$, $X^2_{N\text{xN}}$, and $X^3_{N\text{xN}}$ include right diagonal.
        $X^0_{2N\text{x2N}}$ = Right diagonal
    end if
end if
```

early CU size decision provide the reduction of RDO iteration to decide the optimal CU size.

The intra texture coding is performed with optimal CU size which is decided by early CU size decision algorithm, and RMD calculate the intra prediction mode of 35 mode. On the other hand, if the intra depth coding is performed, the efficient edge detection and the edge classification are calculated. Because the complexity of intra depth coding takes about 5 times more than that of texture coding, the fast decision

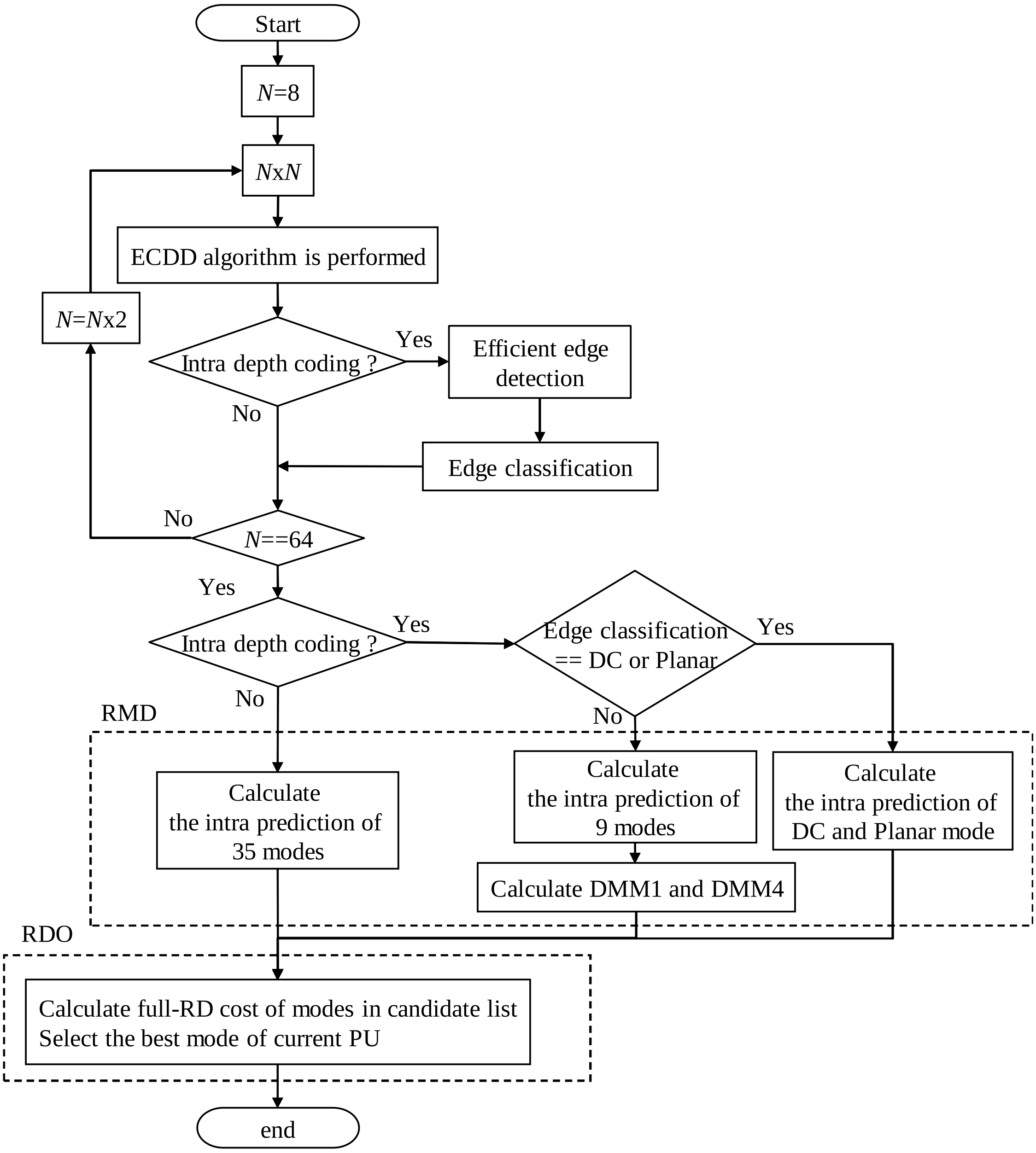

Start
N=8
NxN
ECDD algorithm is performed
N=Nx2
Intra depth coding ?
Yes
No
Efficient edge detection
Edge classification
No
N==64
Yes
Intra depth coding ?
Yes
No
Edge classification == DC or Planar
Yes
No
RMD
Calculate the intra prediction of 35 modes
Calculate the intra prediction of 9 modes
Calculate the intra prediction of DC and Planar mode
Calculate DMM1 and DMM4
RDO
Calculate full-RD cost of modes in candidate list
Select the best mode of current PU
end

Figure 3.18 The processing flow of 3-D intra coding in the proposed algorithm

of the candidate mode number greatly contributes to the complexity reduction of intra depth coding. In particular, compared with previous works, the intra depth prediction mode is decided by a little edge information. The edge classification reduces the RMD process from 35 mode to 9 mode. On the other hand, when the edge classification is DC or Planar, the current CU shows the simple depth map.

In contrast to the intra directional mode, DC and Planar mode also is an important factor for complexity reduction. In current HTM16.0, a fast skip algorithm for DMMs has been adopted. When the best mode in RMD is Planar mode and the pixel variance of current

CU is smaller than a threshold, this indicates the current CU is homogeneous and DMMs should be skipped from the mode decision. Therefore, the Planar and DC mode could be a very important factor in skipping DMMs. In previous work, when DMMs skipped conditions are met, the CU has more than 99% probability to choose Planar and DC mode as the best mode. Therefore, when Planar and DC mode are selected, DMMs are not performed.

Finally, for deciding the best mode of current CU, full-RD cost is calculated from candidate list which is decided by RMD.

3.2.4 Simulation result

The proposed algorithm is implemented in reference software HTM16.0 [42][43]. It follows the common test condition (CTC) [55]. The simulation environment is Intel (R) Core (TM) i7-4770 CPU@3.40GHz with 4 cores, RAM 8.00 GB and Windows 10 Home Edition 64-bit. All test sequences are coded using the all intra configuration for three-view cases. The texture and depth map use the QPs setting to (QP texture, QP depth) = (25,34), (30,39), (35,42) , (40,45). Other encoding parameters remain the same as the CTC. Coding efficiency is measured by BD-BR and BD-PSNR. Time saving (TS) represents the reduced total encoding time, including the texture video coding and the depth video coding. TS is defined as

$$TS\,(\%) = \frac{T_{HTM} - T_{Proposed}}{T_{HTM}} \times 100 \tag{3.30}$$

where T_{HTM} is the encoding time of the original reference software and $T_{proposed}$ is that of the proposed algorithm. The computation complexity reduction (CR) is evaluated with different QP. CR is defined as

$$CR \;=\; \sum_{f=1}^{F}\sum_{j=1}^{J} \frac{\sum_{i=1} BestPU_{(i,j,f)}}{J \times F \times 85} \times 100(\%) \tag{3.31}$$

where $BestCU(i, j, f)$ indicates the ith best CU of the jth CTU of the fth frame of the test sequence. J and F are the total number of CTUs in each frame and the total number of frames of the test video, respectively. 85 represents the number of iteration which is required for the best CU size decision from 64x64 to 8x8 in reference software HTM16.0. Our proposed algorithm reduce the computational complexity in intra texture coding and intra depth coding, respectively.

Table 3.14 Comparison with previous works by in TS and BD-BR

	[51]		[52]		[53]		[54]		Proposed	
Sequences	BD-rate (%)	TS (%)	BD-rate (%)	TS (%)	BD-rate (%)	TS (%)	BD-rate (%)	TS (%)	BD-rate (%)	TS (%)
Ballons	1.2	27.6	0.2	26.2	1.1	32.6	0.0	36.7	-0.1	56.3
Newspaper	1.1	26.6	-2.0	28.9	2.4	45.9	0.1	35.0	-0.1	53.2
Kendo	0.7	26.3	0.1	25.2	1.1	47.9	0.1	33.7	-0.3	59.7
PoznanHall2	1.3	43.7	0.4	32.7	1.8	22.9	0.2	38.4	0.8	47.4
PoznanStreet	1.4	49.1	0.2	33.7	0.4	39.0	0.1	34.5	0.9	56.6
Shark	0.1	33.7	NA	NA	0.4	34.9	0.1	33.6	0.2	55.6
UndoDancer	0.6	49.1	0.3	31.0	0.4	40.6	0.1	32.2	0.8	57.4
GTFly	0.3	45.1	0.1	30.1	0.4	39.2	0.0	34.3	1.3	69.4
Average	0.8	37.6	-0.06	29.7	0.9	37.9	0.1	34.8	0.5	56.9

The results given in Table 3.14 indicate BD-rate comparison by our proposed algorithm and previous works [51]-[54]. Table 3.14 shows that the proposed algorithm reduces TS by about 20% better than in [51]-[54] whereas it achieves better BD-rate as well. In particularly, [53] reduces the total TS by 37.9% at the cost of 0.90% BD-rate increase, on average. [53] can perform CU size decision without data dependencies in HEVC intra-prediction mode, which is a desired characteristic in a hardware design. Because the complexity reduction of the intra depth prediction mode is achieved by our proposed algorithm, our proposed algorithm reduces TS 19.0% compared with the proposed algorithm in [53]. [54] proposed fast CU size and prediction mode decision by using SVM. Compared with our proposed algorithm, the new approach which is proposed by [54] achieves better BD-rate. On the other hand, TS reduction of [54] is lower than our proposed algorithm. Because we consider that DMMs could not be reduced efficiently in [54], our proposed intra depth prediction mode decision algorithm have high efficacy for DMMs.

Moreover, in Table 3.15, the performance of our proposed algorithm is compared with the proposed algorithm in [51] every QP value. The test condition of proposed algorithm is the same as the compared algorithm. The results given in Table 3.15 indicate that the proposed algorithm reduces TS up to 28.7% better than in [51] and our proposed algorithm achieves the complexity reduction for every QP value. According to the above results, we confirm that the computational complexity reduction is achieved with almost no video quality loss.

Table 3.15 Comparison with previous work by TS of proposed intra depth coding under different QPs

Sequences	Proposed			[51]		
	QP34	QP39	QP43	QP34	QP39	QP43
	TS(%)			TS(%)		
Ballons	68.5	51.2	52.9	20.4	20.5	28.2
Newspaper	39.6	50.3	50.4	17.2	15.9	26.1
Kendo	81.3	50.3	53.9	NA	NA	NA
PoznanHall2	66.3	44.1	30.4	NA	NA	NA
PoznanStree	52.3	46.6	72.7	NA	NA	NA
Shark	57.3	60.2	55.4	NA	NA	NA
UndoDancer	48.7	40.6	68.6	43.9	42.5	44.6
GTFly	49.2	72.9	81.8	35.1	33.3	47.0
Average	57.9	52.1	58.3	29.2	28.1	36.5

3.2.5 Conclusion

The focus of this paper is on developing a complexity reduction scheme for 3D-HEVC encoder. The proposed algorithms use fast intra texture and depth coding. Our scheme utilizes the boundary homogeneity to predict the CU sizes of the CTUs of texture and depth coding. To realize the low complexity of CU size decision, our approach notice the boundary homogeneity of every CU size. Moreover, for complexity reduction in the intra depth prediction mode, we proposed the edge classification by using Laplacian filter. The performance of the proposed algorithm was tested on a representative set of video sequences and was compared against the unmodified HTM encoder as well as two of the art complexity reduction schemes and combinations. Performance evaluations show that our proposed algorithms reduce encoding time on average 56.9% and increases BD-rate about 0.5%, compared with HTM 16.0.

Chapter 4 Future extension model of HEVC

4.1 Research motivation

High Efficiency Video Coding (HEVC) was developed by Joint Collaborative Team on Video Coding (JCT-VC). HEVC is to double the coding efficiency as compared to the predecessor H.264/AVC, especially when processing the high-resolution sequences (HD/UHD). However, this increases the computational complexity up by 10 times as compared to H.264/AVC. On the other words, the high computational complexity in HEVC becomes an hardware implementation bottle-neck. In HEVC, recursive CU size decision method occupy most of the computational complexity. Intra coding is particularly an important coding tool adopted in almost all mainstream video compression standard such as MPEG-2, H.264/AVC and HEVC.

Although the quad-tree structure enables each CU to be coded optimally and can greatly improves the encoding efficiency significantly, it imposes significant computational complexity on the encoder during the exhaustive rate-distortion cost calculation of total 85 CUs, where all possible combinations of CU, PU and TU are tried to find the optimal combination. Thus, it is important to find a practical implementation of HEVC to reduce the complexity while maintaining its performance. To overcome this problem, a number of algorithms on accelerating the encoder of HEVC have been proposed to reduce the computational complexity.

To alleviate the intra encoding complexity, many algorithms have been developed for fast intra coding mode decision. The previous work can be classified into three categories.

1. The method of the first category reduces the RDO complexity of intra prediction mode in every CU depth. For example, [56] applied the rough mode decision scheme to reduce the number of candidate prediction modes, which will perform the RDO processing. Zhu simplified the computation of rate and distortion estimation [57][58]. These previous works contribute to the RDO complexity reduction.

2. The methods of the second class dynamically skip the CU depth decision process based on some pre-processing [59][60][61]. Similarly, another algorithms skip the early terminating the CU/PU depth RDO procedure based on the CU depth information of previously coded slices and neighboring CUs [62].
3. The method of the third category shows a fast CU partitioning algorithm using machine learning which has been actively discussed in recent years. Some algorithms decide the optimal CU depth by using the convolutional neural network (CNN) [63][64]. Because the pipeline processing of coding tree unit (CTU) is considered, these previous works is oriented for hardware implementation.

For the hardware encoder design, the first kinds of methods did not reduce the depth of CU/PU. On the other hand, the methods of the second category did not shrink the maximum complexity at the CTU. For example, in literature [9], in the parameter training stage, all CU levels must be searched with the exhaustive RDO. Moreover, in the third category, the inherent drawbacks of [11] induce 4.79% BD-BR increment. In fact, the optimal CU coding modes are determined by not only the edge information, the texture strength, and the quantization step, but also neighboring CU parameter. Therefore, our research focuses on the texture information of neighboring spatial blocks.

We consider that the coding performance is improved by the utilizing the neighboring blocks. However, the input texture of neighboring spatial blocks increase the computation complexity of CNN. In this chapter, the fast CU depth decision is implemented by the optimal CNN architecture, that is specially devised to deal with 16x16 pixel block.

4.2 Analysis of CNN for fast intra coding

4.2.1 Verification of CNN structure

Previous work proposed the hardware-oriented algorithm using CNN for HEVC encoder [64]. This work achieved high computational complexity reduction, and proved easy to implement hardware of CNN. However, the proposed algorithm induced BD-BR increase. Considering the encoding of super-resolution (4K), a more efficient CU decision algorithm by CNN is required for super-resolution encoder. For this reason, to clarify the optimal CNN structure including the convolutional layer (Conv), kernel, and full connection layer (FCL), we evaluate the relationship of the validity and the parameter. The evaluation is performed by using CNN structure of [64], and the block division of 32x32 is judged by single CNN process. CNN structure of [64] is shown as figure 3.17. The structure consists of two Conv, two max pooling, and two FCL. The parameters represent the number of Conv, kernel, and FCL. The sequence of ClassA and ClassB are used as the training sequences.

Table 4.1 represents validation accuracy (Valid_acc) and training accuracy (Train_acc) when training of 20000 epoch is performed. Reference CNN represent simple and small network based on the CNN structure of [64]. To evaluate the accuracy with some conditions, the Conv, kernel, and FCL are added to the reference CNN. The evaluation result shows about 70% Train_acc and 65% Valid_acc. From this evaluation result, it is observed that both accuracy are not greatly affected by the variation of the number of Conv, kernel, and FCL. In other words, in the simple structure, training and validation accuracy have the limit accuracy. Hence, for achieving higher accuracy performance, our approach adopts the CNN structure of the multiple inputs.

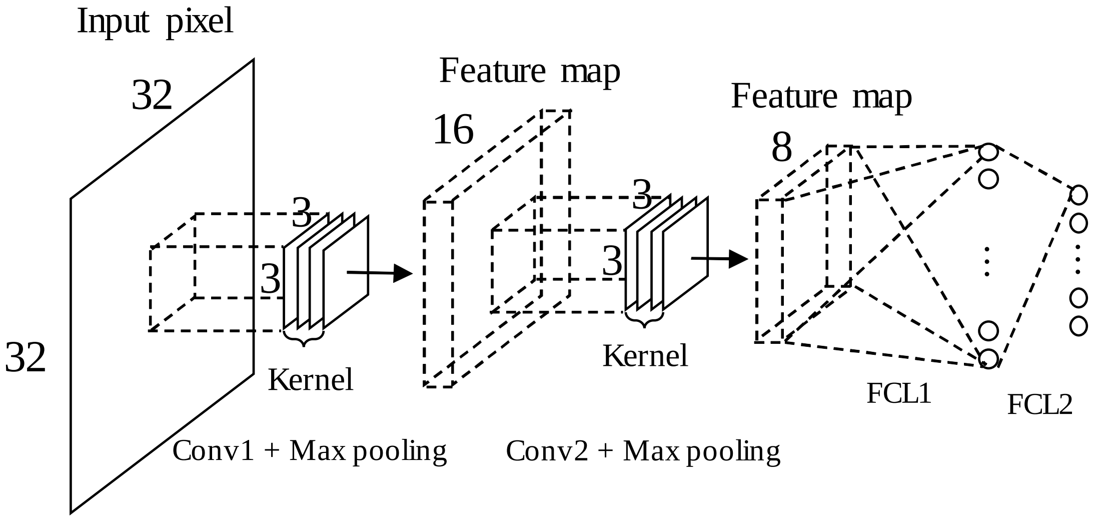


Figure 4.1 Reference CNN structure

In HEVC intra coding, best CU size depend on the complexity of the neighboring

Table 4.1 Analyzing of single input CNN

	Number of parameter			Prediction accuracy	
	Conv	Kernel	FCL	Valid_acc	Train_acc
Default parameter[64]	2	6	2	68.40	71.80
Comparison of Conv	3	6	2	65.00	71.00
	4	6	2	60.40	71.70
Comparison of kernel	2	8	2	49.90	65.10
	2	10	2	65.90	71.00
Comparison of FCL	2	6	3	63.50	70.90
	2	6	4	50.00	55.70

blocks. For this reason, the pixels of the neighboring blocks are important for high prediction accuracy. Figure 4.2 shows the block position which is used as input to our proposed CNN. In our approach, only neighboring blocks are used for the input of CNN. Moreover, the prediction of division pattern in 32x32 block have high complexity. Considering the improvement of the division accuracy, the reduction of block division pattern is required. Therefore, our proposed CNN use 16x16 block as input texture. The CNN structure of multiple inputs that we considered from this evaluation result is shown in figure 4.3. To clarify the structure of more efficiency multiple inputs CNN, in the next subsection, the evaluation of the number of input and parameter is performed by using figure 4.3.

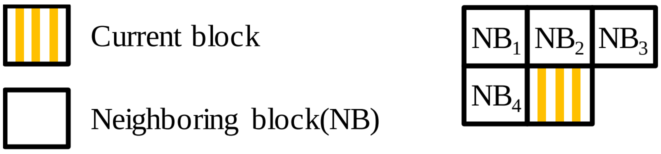


Figure 4.2 Mapping neighboring block and current block

4.2.2 Evaluation of multiple inputs CNN

With the support of powerful computational devices, many deep learning networks become deeper. For example, a deep residual network is proposed for image recognition [65]. However, the deeper network model does not necessarily give the

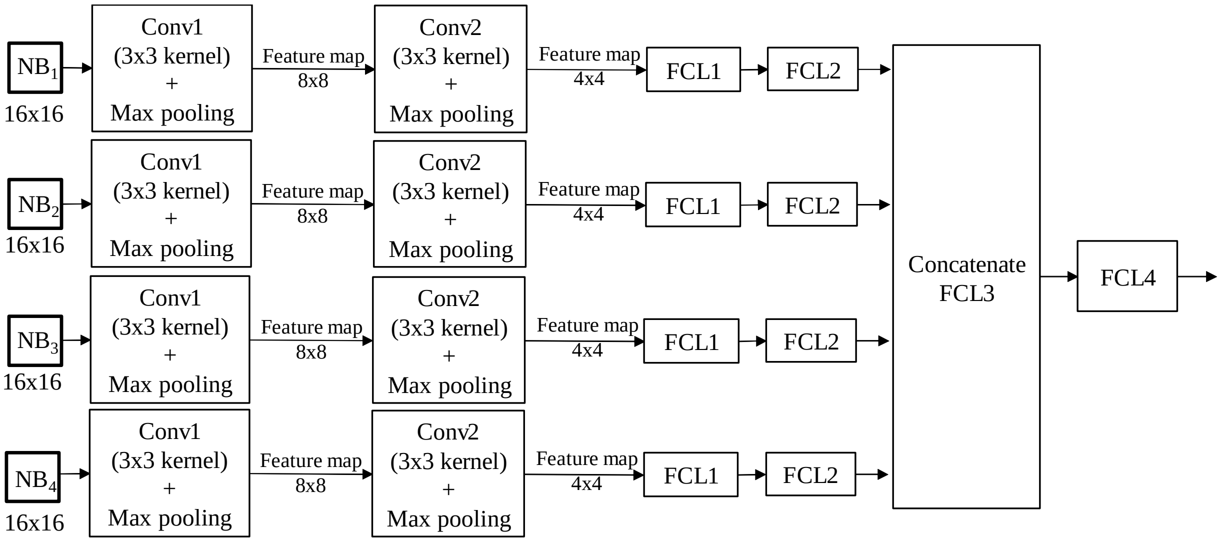


Figure 4.3 Structure of 4-inputs CNN

high accuracy. Therefore, for identifying the most suitable multiple inputs CNN, we evaluate the prediction accuracy according to the different of the number of inputs. Variation of the number of the neighboring block and the number of Conv, kernel, FCL were evaluated with the sequence of ClassA and ClassB, as shown in Table 4.2. Table 4.2 shows that the increasing of the neighboring block and kernel lead to improvement of accuracy. It is clear that the extraction of feature map by kernel is important for block division. On the other hands, about the variation of Conv and FCL, these parameters had little change compared to kernel variation.

Table 4.2 Train and validation accuracy (%) evaluation of the neighboring block and the parameter variation.

	Number of parameter			Number of input			
				NB2, NB4		NB1, NB2, NB3, NB4	
	Conv	Kernel	FCL	Valid_acc	Train_acc	Valid_acc	Train_acc
Default parameter	2	6	2	68.40	73.40	69.40	76.80
Comparison of Conv	3	6	2	69.10	72.80	70.40	74.80
Comparison	2	8	2	73.20	78.10	75.80	80.20
of kernel	2	10	2	79.90	84.50	84.40	89.20
Comparison	2	6	3	66.80	68.80	67.20	71.80
of FCL	2	6	4	65.40	67.80	66.40	68.80

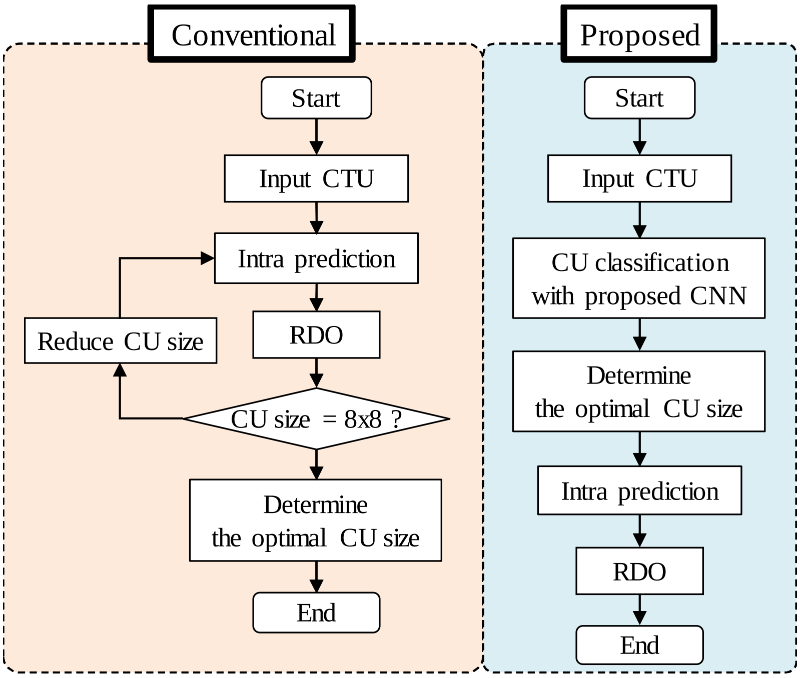


Figure 4.4 Comparison of conventional flowchart and the proposed flowchart.

4.3 Proposed algorithm

A module with our CNN model is implemented and embedded in HM16.7 encoder software before intra prediction. The CNN classifier outputs the optimal CTU division information. The detail of flowchart is shown in figure 4.4. Compared with the conventional encoding processing, the encoding process using our proposed algorithm is not induce the many iteration for determining the optimal CU depth. The CU classification algorithm will be helpful to the computational complexity reduction of intra encoding. This means that the hardware area of RDO process in intra coding mode can be reduced.

From evaluations previous section, our approach supplies the best performance condition to CNN model. In Table 4.2, the comparison of Conv, Kernel, FCL are evaluated. Obviously, the number of input affect the prediction accuracy. Regarding Conv and FCL parameter, the prediction accuracy improved a little. As a possible reason for that increasing the Conv in 16x16 block is ineffective and many parameters in FCL increase the complexity of the prediction accuracy. On the other hand, the number of kernel increases the effective parameter for classification. Therefore, our

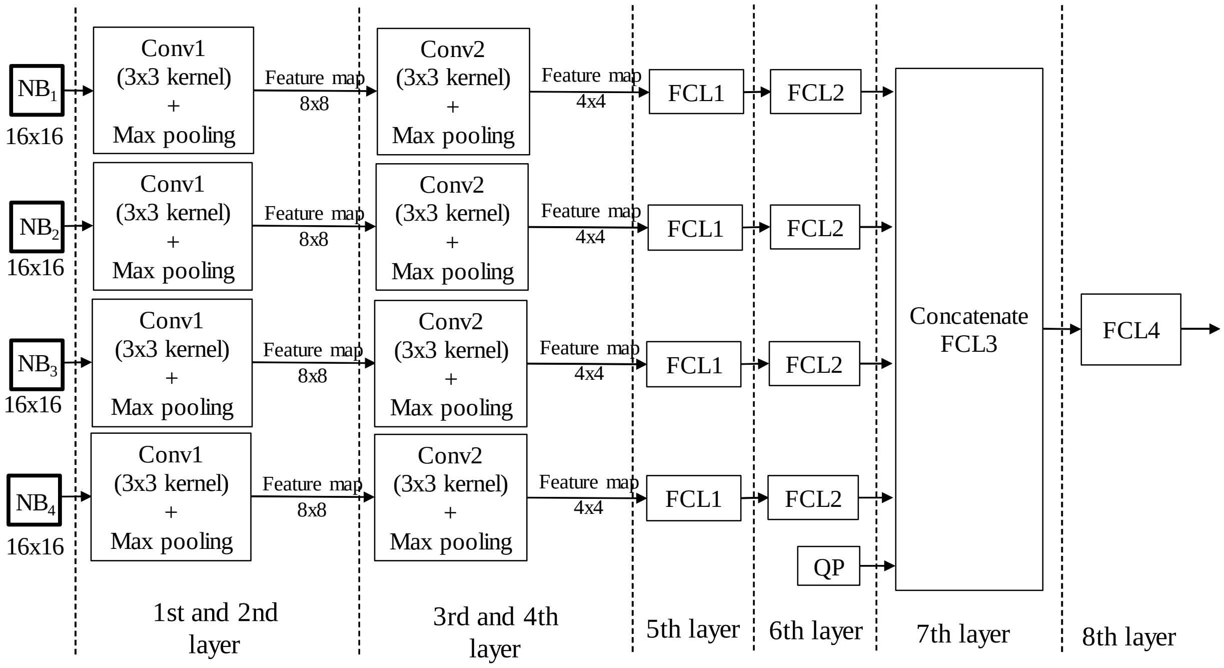


Figure 4.5 Proposed CNN structure

proposed CNN consist of the multiple inputs using neighboring block, two Convs, two FCLs, and ten kernels, as shown in figure 4.5.

The input of CNN is the block partition patterns from 64x64 to 8x8 which are converted to 16x16 block. The first layer is a convolutional layer with ten kernels. Each neuron is connected to a 3x3 receptive field in the input. The size of the feature map is 8x8 and the convolution calculation is performed with zero padding mode. The kernels in this layer are deemed as feature extractors. The second layer performs the max pooling. Similarly, the third and fourth layer perform the convolution and max pooling. The fifth and sixth layers use FCL which the parameter is 256 and 64 to each input. The seventh layer concatenate FCL of each input with 256 parameters. The eighth layers perform FCL which the parameter is 64. The output layer uses cross entropy units.

4.4 Simulation result

For all of the evaluations, the coding structure which is set to all intra mode is used. The simulation environment is Intel (R) Core (TM) i7-4770 CPU 3.40GHz with 4 cores, RAM 8.00 GB and Windows 10 Home Edition 64-bit. Several test sequences (30 frames) with picture size from Class 4K to Class B are used. The computation complexity reduction (CR) is evaluated with QP from 22 to 37. CR is defined as

$$CR = \sum_{f=1}^{F}\sum_{j=1}^{J}\frac{\sum_{i=1} BestCU_{(i,j,f)}}{J \times F \times 85} \times 100(\%) \tag{4.1}$$

where $BestCU_{(i,j,f)}$ indicates the ith best CU of the jth CTU of the fth frame of the test sequence. J and F are the total number of CTUs in each frame and total number of frames of the test video, respectively. 85 is represented by the number of iteration which is required for the best CU size decision from 64×64 to 8×8 in reference software HM16.7.

The results of our experiment are summarized in Table 4.3. The coding performance comparisons between the proposed algorithm and the original HM16.7 are shown in Table 4.3. The proposed algorithm shows a consistent gain in encoding time saving for all sequences with the least gain of 60.6% in Rollercoaster and the most gain of 75.8% in “Kimono”. For all sequences, the proposed algorithm can save 66.7% encoding time and 70.1% complexity reduction.

In Table 4.4, the time reduction percentage compared to previous work [67] is shown, the impact on the bit-rate (bjontegaard delta rate (BD-BR)), and the video quality in terms of PSNR (BD-PSNR in dB) [68]. Table 4.4 shows that the proposed algorithm reduces TS by about 1.2% better than in [67]. Additionally, BD-BR -2.58% and BD-PSNR 0.09dB are improved by our proposed algorithm, respectively. In particularly, our approach have an effect on the complexity texture such as “Traffic” and “BasketballDrive”.

Moreover, in order to confirm feasibility of the hardware implementation, the total number of parameter required for the machine learning module is verified. From Table 4.5, 19538 parameters are required for the proposed architecture. The proposed architecture implemented with the number of parameter below the upper limit of the machine learning module that is possible with the current mobile terminal device. Compared with the with other papers in the number of parameter, time saving, BD-BR and BD-PSNR in the previous work using the machine learning

Table 4.3 Result of proposed algorithm compared to HM16.0

Class	Sequences	CR(%)	TS(%)	BD-rate (piecewise cubic)		
				Y(%)	U(%)	V(%)
Class 4K	CampfireParty	69.5	62.0	1.5	1.6	1.7
	CatRobot	78.3	62.9	1.9	2.1	2.4
	DaylightRoad	62.8	62.4	2.6	2.6	2.0
	Drums100	77.9	66.1	1.6	1.9	0.9
	TrafficFlow	65.2	61.6	2.3	2.1	2.8
	Tango	68.2	62.7	1.7	1.6	1.9
	ToddlerFountain	70.3	62.5	1.3	1.4	1.7
	Rollercoaster	66.1	60.6	2.3	2.2	2.2
	Average	69.8	62.6	1.9	1.9	2.0
Class A	Traffic	70.4	68.7	1.4	1.9	1.8
	PeopleOnStreet	72.8	66.5	2.4	1.9	1.8
	Nebuta	72.5	67.1	1.5	1.7	1.8
	SteamLocomotive	71.4	64.1	1.8	0.6	1.1
	Average	71.8	66.6	1.8	1.5	1.8
Class B	BasketballDrive	68.3	72.1	1.8	1.3	0.8
	BQTerrace	66.1	73.3	1.8	0.4	0.1
	Cactus	68.9	71.8	2.0	1.6	1.7
	Kimono	72.5	75.8	1.7	1.5	1.7
	ParkScene	71.3	73.7	1.2	0.1	0.7
	Average	69.4	73.3	1.7	1.0	1.2
	Average	70.1	66.7	1.8	1.6	1.6

module, it is clear that the proposed architecture achieves high efficiency encoding with few parameters.

Table 4.4 Comparison with other paper in time saving, BD-BR and BD-PSNR

	Previous work[28]			Proposed algorithm			[28] vs Proposed algorithm		
Sequences	TS (%)	BD-BR (%)	BD-PSNR (dB)	TS (%)	BD-BR (%)	BD-PSNR (dB)	ΔTS (%)	ΔBD-BR (%)	ΔBD-PSNR(dB)
PeopleOnStreet	74.6	5.24	-0.26	72.8	3.23	-0.13	-1.8	-2.01	0.13
Traffic	73.4	5.01	-0.24	70.4	2.11	-0.09	-3.0	-2.90	0.15
BasketballDrive	76.1	5.52	-0.14	76.3	1.93	-0.08	0.2	-3.59	0.06
BQTerrace	72.3	4.03	-0.20	73.1	1.92	-0.08	0.8	-2.11	0.12
Cactus	77.5	4.72	-0.16	74.9	2.16	-0.18	-2.6	-2.56	-0.02
Kimono	62.6	3.64	-0.12	75.5	1.56	-0.06	12.9	-2.08	0.06
ParkScene	72.0	3.97	-0.16	74.3	1.13	-0.05	2.3	-2.84	0.11
Average	72.6	4.59	-0.18	73.9	2.01	-0.10	1.3	-2.58	0.09

Table 4.5 Comparison with other paper in the number of parameter, time saving, BD-BR and BD-PSNR

	Number of parameter	TS	BD-BR	BD-PSNR
		(%)	(%)	(dB)
Proposed architecture	19538	73.2	2.21	-0.10
Liu's architecture[64]	700	73.1	4.75	-0.20
Mai's architecture[69]	757929	70.7	2.36	-0.10
MobileNet[70]	4200000	N/A	N/A	N/A

Chapter 5 Overall conclusion and future work

Scalable coding enables single channel transmission, which can serve many different quality levels and resolutions. This allows for cost effective universal access of digital media by a variety of playback devices and different bandwidth requirements (ranging from TV displays and computers to tablets and smart phones), an attractive proposition for broadcasters and end-users alike. In this work we presented coding methods that significantly reduce the complexity of the original scalable standards, leading to simpler hardware and software implementations, which have the potential to accelerate the wide adoption of these standards. Our methods are implemented on MPEG's SHVC reference software model (SHM) and 3D-HEVC reference software model (3D-HTM). As such, our contributions are ready to be used by the industry. Note that all our methods are introduced at the encoder side with the decoder left untouched. The performance of our methods is compared with that of the unmodified reference encoding in terms of execution time and compression performance. Related contributions were submitted and presented at the consumer electronics society meetings. In particular, we develop a complexity reduction schemes for spatial SHVC (Chapter 3).

In other work, we proposed several complexity reduction methods for 3D-HEVC, the latest Multiview/3D compression standard. Lately, TV manufacturers are placing their hopes for the future of 3D TV on the so-called "glasses-free" 3D TV technology, where "autostereoscopic" displays show multiple views of 3D (ranging from 8 to more than 100) without the need for glasses. Autostereoscopic glasses-free displays and 360 video applications, which are just emerging require a large number of views that translate to large amounts of data and of course challenging issues in relation to transmission and storage. Our proposed methods were utilized to reduce complexity reduction for the case of having 3D streams with 2 views and their corresponding depth maps, but they can be easily adopted for the auto-stereoscopic and 360-degree video applications. Another attractive application for these methods is the case of 3D virtual reality and augmented reality where encoding delays for real-time applications will be a challenge. In such applications, for instance, any movement of the user wearing augmented reality glasses in the center of a scene, will require the transmission of a new video scene to the glasses. News and events transmitted from remote scenes to the main station by reporters and crews using portable cameras also face power consumption challenges. Our methods may directly address these challenges in cases of scalable streaming and multiview broadcasting.

Chapter 4 shows that the future extension model is reviewed by using CNN and HEVC. Considering the society that realize AI-IoT in the future, machine learning devices are implemented to various products. Under this environment, machine learning devices can be effectively utilized. For the reasons, the object of the future extension model is to new codec model implementation using the machine learning. The new encoding model using the proposed CNN reduces the computation complexity significantly. The proposed CNN structure are composed of eight layers and 19538 parameters. The feasibility of the proposed architecture is confirmed by comparing the number of parameter with previous works. However, our proposed CNN architecture may impact to increase of hardware area because the CNN structure of multiple inputs requires the parallel processing. Additionally, the utilization of much kernels needs to reserve many parameters. Therefore, in our CNN structure, a large memory area is required. To solve these problems, we need to consider the reduction of number of input pixel and kernel parameter in our future work. As major goal of our future work, we expect that a new codec system will be established by using the machine learning.

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, July 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. June 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. Jan. 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. Jan. 2016.

[6] 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, Dec. 2012.

[7] V. Sze, M. Budagavi, G.J. Sullivan, “High efficiency video coding (HEVC),” document Springer, 2014.

[8] 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, Oct. 2012.

[9] 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, DEC. 2012.

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

[11] A. H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," IEEE Trans. Circuits Syst. Video Technol., vol. 17, no.9, pp. 1103-1120, Sep. 2007.

[12] H. R. Tohidypour, M. T. Pourazad, and P. Nasiopoulos, "Content adaptive complexity reduction scheme for quality/fidelity scalable HEVC," in Proc. Int. Conf. Acoustics, Speech, Signal Process. (ICASSP), pp. 1744-1748, May. 2013.

[13] I. K. Kim, K. McCann, K. Sugimoto, B. Bross, and W. J. Han, "High efficiency video coding (HEVC) test model 10 (HM10) encoder description," document JCTVC-L1002, Jan. 2013.

[14] 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, Jun. 2016.

[15] J. Chen, J. Boyce, Y. Ye, M. Hannuksela, and G. Barroux, "Scalable HEVC (SHVC) Test Model 11 (SHM 11)," document N15778, Oct. 2015.

[16] Z. Zhao, J. Si, and J. Ostermann, "Inter-layer intra prediction mode coding for the scalable extension of HEVC," document JCTVC-K0238, Oct. 2012.

[17] W. Jiang, H. Ma, and Y. Chen, "Gradient based fast mode decision algorithm for intra prediction in HEVC," in Proc. Int. Conf. Consum. Electron., Commun. Net. (CECNet), pp. 1836-1840, Apr. 2012.

[18] G. Chen, Z. Liu, T. Ikenaga, and D. Wang, "Fast HEVC intra mode decision using matching edge detector and kernel density estimation alike histogram generation," in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), pp. 53-56, May 2013.

[19] L. Shen, Z. Zhang, and Z. Liu, "Effective CU size decision for HEVC intra coding," IEEE Trans. Image Proc., Vol. 23, pp. 4232-4241, Oct. 2014.

[20] S. Na, W. Lee, and K. Yoo, "Edge-based fast mode decision algorithm for intra prediction in HEVC," in Proc. IEEE Int. Conf. Consum. Electron. (ICCE), pp.11-14, Jan. 2014.

[21] A. Aguilar-González, M. Arias-Estrada, M. Pérez-Patricio, and J. L. Camas-Anzueto, "An FPGA 2D-convolution unit based on the CAPH language," J.

Real-Time Image Process., Springer-Verlag Berlin Heidelberg, pp 1?15, Oct. 2015.

[22] M. Ramezanpour, and F. Zargari, "Fast HEVC I-frame coding based on strength of dominant direction of CUs," J. Real-Time Image, Springer-Verlag Berlin Heidelberg, Volume 12, Issue 2, pp 397-406, Aug. 2016.

[23] J. Zhu, Z. Liu, D. Wang, Q. Han, and Y. Song, "HDTV1080p HEVC Intra encoder with source texture based CU/PU mode predecision," in Proc. Design Aut. Conf. (ASP-DAC), pp. 367-372, Jan. 2014.

[24] X. Huang, H. Jia, B. Cai, C. Zhu, J. Liu, M. Yang, D. Xie, and W. Gao, "Fast algorithms and VLSI architecture design for HEVC intra-mode decision," J. Real-Time Image, Special issue paper, pp.1-18, Dec. 2015.

[25] S.-F. Tsai, C.-T. Li, H.-H. Chen, P.-K. Tsung, K.-Y. Chen, and L.-G. Chen, "A 1062Mpixels/s 8192x4320p high efficiency video coding (H.265) encoder chip," in Proc. Symp. on VLSI Circuits (VLSIC), pp. C188-C189, June 2013.

[26] H. R. Tohidypour, "Adaptive search range method for spatial scalable HEVC," in Proc. IEEE Int. Conf. Consum. Electron. (ICCE) , pp.191-192, Jan. 2014.

[27] X. Zuo, "Fast mode decision method for all intra spatial scalability in SHVC," in Proc. IEEE Visual Commu. Image Process. Conf. (VCIP), pp.394-397, Dec. 2014.

[28] H. R. Tohidypour, M. T. Pourazad, and P. Nasiopoulos, "Probabilistic approach for predicting the size of coding units in the quad-tree structure of the quality and spatial scalable HEVC," IEEE Trans. Multimedia, vol. 18, np. 2, pp.182-185, Feb. 2016.

[29] T. Katayama, W. Shi, T. Song and, T. Shimamoto, "Early depth determination algorithm for enhancement layer intra coding of SHVC," in Proc. IEEE Int. Tec. Conf. (TENCON), pp.3083-3086, Singapore, Nov. 2016.

[30] L. Zhao, X. Fan, S. Ma, D. Zhao, "Fast intra-encoding algorithm for high efficiency video coding," Signal Process. Image Commu., vol. 29, np. 9, pp. 935-944, June 2014.

[31] B. Li, H. Li, L.Li, and J. Zhang, "λ domain rate control algorithm for high

efficiency video coding," IEEE Trans. Image Process., pp.3841-3854, Sept. 2014.

[32] Ultra Video Group test sequences [Online]. Available:http://ultravideo.cs.tut.fi/#testsequences.

[33] V. Seregin and Y. He "Common SHM test conditions and software reference configurations," document JCTVC-Q1009, Mar. 2014.

[34] F. Bossen, "Common test conditions and software reference configurations," document JCTVCL-1100, Jan. 2013.

[35] HEVC test model 16.7. [Online]. Available:https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.7/

[36] SHM test model 11.0. [Online]. Available:https://hevc.hhi.fraunhofer.de/svn/svn_SHVCSoftware/tags/SHM-11.0/

[37] S. Pateax, "An excel add-in for computing bjontegaard metric and its evolution," document VCEG-AE07, Apr. 2007.

[38] K. Muller, H. Schwarz, D. Marpe et al., "3D high-efficiency video coding for multi-view video and depth data," IEEE Trans. Image Process., Vol. 22, Issue. 9, pp. 3366-3378, May 2013.

[39] P. Kauff, N. Atzpadin, C. Fehn et al., "Depth map creation and image-based rendering for advanced 3DTV services providing interoperability and scalability," Signal Process. Image Commu., vol. 22, Issue. 2, pp. 217-234, Feb. 2007.

[40] Q. Zhang, P. An, Y. Zhang, L. Shen, and Z. Zhang, "Efficient depth map compression for view rendering in 3D video," Imaging Science Journal, Vol. 61, Issue. 4, pp. 385-395, Nov. 2013.

[41] G. Tech, Y. Chen, K. Muller et al., "Overview of the multiview and 3D extensions of high efficiency video coding," IEEE Trans. Circuits Syst. Video Technol., Vol. 26, Issue. 1, pp.35-49, Sep. 2016.

[42] Y. Chen, G. Tech, K. Wegner, and S. Yea, "Test Model 11 of 3DHEVC and MV-HEVC," document JCT3V-K1003 ISO/IEC JTC1/SC29/WG11, MPEG/m36142, Feb. 2015.

[43] K. Suehring and K. Sharman, HEVC reference software, date:[2013], online available at: https://hevc.hhi.fraunhofer.de/trac/3dhevc/browser/3DVCSoftware/tags/HTM-16.0.

[44] C. Fu, W. Su, and S. Tsang, “Fast wedelet pattern decision for DMM in 3D-HEVC,” in Proc. IEEE Int. Conf. Digital Signal Process. (DSP), pp.477-481, July 2015.

[45] G. Sanchez, R. Cataldo, R. Fernandes et al., “3D-HEVC depth maps intra prediction complexity analysis,” in Proc. IEEE Int. Conf. Elect. Circuits and Sys. (ICECS), pp. 348-351, June 2016.

[46] Z. Gu, J. Zheng, N. Ling, and P. Zhang, “Simplified depth intra mode selection for 3D video compression,” in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), pp.1110-1113, June 2014.

[47] Z. Gu, J. Zheng, N. Ling, and P. Zhang, “Fast bipartition mode selection for 3D HEVC depth intra coding,” in Proc. IEEE Int. Conf. Multimedia and Expo (ICME), pp.1-6, July 2014.

[48] C. Park, “Edge-based intramode selection for depth-map coding in 3D-HEVC,” IEEE Trans. Image Process., Vol.24, Issue. 1, pp.155-162, Nov. 2015.

[49] T.L.da Silva, L.V.Agostini, L.A.da Silva Cruz, “Complexity reduction of depth intra coding for 3D video extension of HEVC,” in Proc. IEEE Visual Commu. Image Process. Conf. (VCIP), pp.229-232, Dec. 2014.

[50] R. Guo, G. He, Y. Li, and K. Wang, “Fast algorithm for prediction unit and mode decisions of intra depth coding in 3D-HEVC,” in Proc. IEEE Int. Conf. Image Process. (ICIP), pp.1121-1125, Sept. 2016.

[51] K. Peng, J. Chiang and W. Lie, “Low complexity depth intra coding combining fast intra mode and fast CU size decision in 3D-HEVC,” in Proc. IEEE Int. Conf. Image Process. (ICIP), pp.2381-8549, Sept. 2016.

[52] G. Sanchez, M. Saldanha, G. Balota, B. Zatt, M. Porto, and L. Agostini, “Complexity reduction for 3D-HEVC depth maps intra-frame prediction using simplified edge detector algorithm,” in Proc. IEEE Int. Conf. Image Process. (ICIP),

pp. 3209-3213. Oct. 2014.

[53] G. Sanchez, M. Saldanha, M. Porto et al., “Real-time simplified edge detector architecture for 3D-HEVC depth maps coding,” in Proc. IEEE Int. Conf. on Elect., Circuits and Sys. (ICECS), pp. 352-355, Dec. 2016.

[54] H. Zheng, J. Zhu, H. Zeng et al., “Low complexity depth intra coding in 3D-HEVC based on depth classification,” in Proc. IEEE Visual Commu. Image Process. Conf. (VCIP), pp. 1-4, Nov. 2016.

[55] K. Muller and A.Vetro, “Common test conditions of 3DV core experiments,” document JCT3V-G1100, ITU-T SG 16 WP 3 and ISO/IEC JTC1/SC 29/WG 11, Oct. 2014.

[56] S. Ma, S. Wang, S. Wang, L. Zhao, Q. Yu, and W. Gao, “Low complexity rate distortion optimization for HEVC,” in Proc. Data Comp. Conf. (DCC), pp.73-82, Mar. 2013.

[57] J. Zhu, Z. Liu, D. Wang, Q. Han, and Y. Song, “Fast prediction mode decision with hadamard transform based rate-distortion cost estimation for HEVC intra coding,” in Proc. IEEE Int. Conf. Image Process. (ICIP), pp.1977-1981, Sep. 2013.

[58] Z. Liu, S. Guo, and D. Wang, “Binary classification based linear rate estimation model for HEVC RDO,” in Proc. IEEE Int. Conf. Image Process. (ICIP), pp.3676-3680, Sep. 2014.

[59] H. Zhang and Z. Ma, “Fast intra mode decision for High Efficiency Video Coding (HEVC),” IEEE Trans. Circuits Syst. Video Technol., vol.24,pp.660-668, Nov. 2012.

[60] Y. Zhang, Z. Li, and B. Li, “Gradient-based fast decision for intra prediction in HEVC,” in Proc. IEEE Visual Commu. Image Process. Conf. (VCIP), pp.1-6, Jan. 2012.

[61] B. Min and R.C.C. Cheung, “A fast cu size decision algorithm for the HEVC intra encoder,” IEEE Trans Circuits Syst. Video Technol., vol.25, pp.892-896, Oct. 2015.

[62] N. Hu and E.-H. Yang, “Fast mode selection for HEVC intra frame coding with

entropy coding refinement based on transparent composite model," IEEE Trans Circuits Syst. Video Technol., vol.25, pp.1521-1532, Jan. 2015.

[63] X. Yu, Z. Liu, J. Liu, Y. Gao, and D. Wang, "VLSI friendly fast CU/PU mode decision for HEVC intra encoding: Leveraging convolution neural network," in Proc. IEEE Int. Conf. Image Process. (ICIP), pp.1285-1289, Sept. 2015.

[64] Z. Liu, X. Yu, S. Chen, D. Wang, "CNN oriented fast HEVC intra CU mode decision," in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), pp.2270-2273, Aug. 2016.

[65] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," arXiv preprint arXiv:1603.05027, 2016.

[66] F. Bossen, "Common HM test conditions and software reference configurations," document. JCTVC-L1100, Apr. 2013.

[67] Z. Liu, X. Yu, Y. Gao, S. Chen, X. Ji, D. Wang, "CU partition mode decision for HEVC hardwired intra encoder using convolution neural network," IEEE Trans. Image Process., vol.25, pp.5088-5103, Aug. 2016.

[68] S. Pateax, "An excel add-in for computing Bjontegaard metric and its evolution," document. VCEG-AE07, Apr. 2007.

[69] X. Mai et al. , " Reducing complexity of HEVC: a deep learning approach, " arXiv:1710.01218v3, Mar. 2018.

[70] Howard et al. , " MobileNets: efficient convolutional neural networks for mobile vision applicationsar , " Xiv:1704.04861v1, Apr. 2017.

Publications

Main papers

1. Takafumi Katayama, Tian song, Wen Shi, Xiantao Jiang, and Takashi Shimamoto : Fast CU Determination Algorithm based on Convolutional Neural Network for HEVC, International Journal of Machine Learning and Computing, Vol. 8, No. 5, October 2018.
2. Takafumi Katayama, Hiroki Tanaka, Tian Song and Takashi Shimamoto : Saliency-Detection-Based Quantization Parameter Setting Method for HEVC, IEEJ Transactions on Electronics Information and Systems, Vol.138, No.10, 1185-1186, 2018 年.
3. Takafumi Katayama, Tian Song, Wen Shi, Xiantao Jiang and Takashi Shimamoto : Boundary Correlation-based Intracoding for SHVC Algorithm and its Efficient VLSI Architecture, Journal of Real-Time Image Processing, Vol. 15, Issue 1, 107-122, June 2018.
4. Takafumi Katayama, Tian Song, Wen Shi, Gen Fujita, Xiantao Jiang and Takashi Shimamoto : Hardware Oriented Low-Complexity Intra Coding Algorithm for SHVC, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Vol.E100-A, No.12, 2936-2947, Dec. 2017.
5. Takafumi Katayama, Tian Song, Wen Shi, Xiantao Jiang and Takashi Shimamoto : Fast Edge Detection and Early Depth Decision for Intra Coding of 3D-HEVC, International Journal of Advances in Computer and Electronics Engineering, Vol.2, No.7, 11-20, July 2017.

Sub papers

1. Xiantao Jiang, Xiaofeng Wang, Tian Song, Wen Shi, Takafumi Katayama, Takashi Shimamoto and Jenq-Shiou Leu : An Efficient Complexity Reduction Algorithm for CU Size Decision in HEVC, International Journal of Innovative Computing, Information and Control, Vol.14, No.1, 309-322, Feb. 2018.

2. Wen Shi, Tian Song, Takafumi Katayama, Xiantao Jiang and Takashi Shimamoto :Hardware Implementation-Oriented Fast Intra-Coding Based on Downsampling Information for HEVC, Journal of Real-Time Image Processing, Vol. 15, Issue 1, 57-71, Mar. 2017.

3. Xiantao Jiang, Tian Song, Wen Shi, Takafumi Katayama, Takashi Shimamoto and Lisheng Wang : Fast Coding Unit Size Decision Based on Probabilistic Graphical Model in High Efficiency Video Coding Inter Prediction, IEICE Transactions on Information and Systems, Vol.E99-D, No.11, 2836-2839, Dec. 2016.

4. Takaaki Hamamoto, Tian Song, Takafumi Katayama and Takashi Shimamoto : Complexity Reduction Algorithm for Hierarchical B-Picture of H.264/SVC, International Journal of Innovative Computing, Information and Control, Vol.7, No.1, 445-457, 2011.

5. Kentaro Takei, Takafumi Katayama, Tian Song and Takashi Shimamoto : Complexity Reduction Algorithm for Enhancement Layer of H.264/SVC, ICIC Express Letters, Vol.4, No.5(B), 1965-1972, 2010.

6. Takaaki Hamamoto, Takafumi Katayama, Tian Song and Takashi Shimamoto : Low Complexity Mode Selection Method for Hierarchical B-picture of H.264/SVC, ICIC Express Letters, Vol.3, No.4(A), 1179-1184, 2009.

Conference papers

1. Takafumi Katayama, Tian Song and Takashi Shimamoto : QP Adaptation Algorithm for Low Complexity HEVC based on a CNN-Generated Header Bits Map, Proceedings of IEEE 8th International Conference of Consumer Electronics in Berlin (ICCE-Berlin), 1-5, Berlin, Sep. 2018.

2. Takafumi Katayama, Koki Tamura, Tian Song and Takashi Shimamoto : Efficient Object Detection Algorithm using Encoded Video Information, Proceedings of Taiwan and Japan Conference on Circuits and Systems (TJCAS2018), No.3AM3A-1, Taichung, Taiwan, Aug. 2018.

3. Koki Tamura, Takafumi Katayama, Tian Song and Takashi Shimamoto :

Multi-line Intra Prediction for Inter-Layer Reference in SHVC, Proceedings of Taiwan and Japan Conference on Circuits and Systems (TJCAS2018), No.3AM3C-2, Taichung, Taiwan, Aug. 2018.

4. Shota Yusa, Takafumi Katayama, Tian Song and Takashi Shimamoto : Fast PU Size Decision Algorithm Using RD-Cost and Depth Dispersion for Depth Intra Coding of 3D-HEVC, Proceedings of Taiwan and Japan Conference on Circuits and Systems (TJCAS2018), No.3AM3C-1, Taichung, Taiwan, Aug. 2018.

5. Kunihito Hatai, Ryota Fujiki, Akinori Sanda, Wen Shi, Takafumi Katayama, Tian Song and Takashi Shimamoto : Learning Library Extension for Sea Cucumber Recognition using GANs, Proceedings of International Technical Conference on Circuits/Systems, Computers and Communications(ITC-CSCC2018), No.CP-02-153, 1-2, Bangkok, Jul. 2018.

6. Akinori Sanda, Kunihito Hatai, Ryota Fujiki, Takafumi Katayama, Tian Song and Takashi Shimamoto : Fast Object Detection System Based on the Embedded Information in the Bitstream, Proceedings of International Technical Conference on Circuits/Systems, Computers and Communications(ITC-CSCC2018), No.CP-02-145, 1-4, Bangkok, Jul. 2018.

7. Takafumi Katayama, Kazuki Kuroda, Tian Song, and Takashi Shimamoto : Low-Complexity Intra Coding Algorithm based on Convolutional Neural Network for HEVC, International Conference on Information and Computer Technologies (ICICT 2018), Northern Illinois University (NIU) DeKalb, USA, Mar. 2018.

8. Takafumi Katayama, Kazuki Kuroda, Tian Song, Jenq-Shiou Leu and Takashi Shimamoto : Multi-Input CNN-based Low-Complexity Intra Coding Algorithm for HEVC, Proceedings of 2nd International Forum on Advanced Technologies (IFAT2018), Tokushima, Mar. 2018.

9. Takafumi Katayama, Wen Shi, Tian Song and Takashi Shimamoto : Pixel-based Fast CU Depth Decision Algorithm with Edge Strength for HEVC, Proceedings of 2018 IEEE International Conference on Consumer Electronics

(ICCE), Las Vegas，Jan. 2018.

10. Wen Shi, Takafumi Katayama, Tian Song and Takashi Shimamoto : Efficient Intra Prediction Based on Adaptive Downsampling Signal for Parallel HEVC Encoding, Proceedings of IEEE International Conference on Consumer Electronics(ICCE2018), 621-624, Las Vegas, Jan. 2018.

11. Yoshiki Ito, Tian Song, Wen Shi, Takafumi Katayama and Takashi Shimamoto : Hardware-oriented Low Complexity Motion Estimation for HEVC, Proceedings of IEEE International Conference on Consumer Electronics(ICCE2018), 438-442, Las Vegas, Jan. 2018.

12. Kazuki Kuroda, Takafumi Katayama, Tian Song and Takashi Shimamoto : Early Mode Selection of High Resolution for HEVC Base on Bits-Mapping, Proceedings of IEEE International Conference on Consumer Electronics(ICCE2018), 428-433, Las Vegas, Jan. 2018.

13. Jiang Xiantao, Wang XiaoFeng, Yang Yadong, Tian Song, Shi Wen and Takafumi Katayama : A Coding Efficiency Improvement Algorithm for Future Video Coding, IFIP International Federation for Information Processing 2017, 279-287, Shanghai, Oct. 2017.

14. Koki Tamura, Takafumi Katayama, Wen Shi, Tian Song and Takashi Shimamoto : Coding Efficiency Improvement Algorithm for Inter-Layer Reference Prediction in SHVC, Proceedings of International Technical Conference on Circuits/Systems, Computers and Communications(ITC-CSCC2017), Busan, Jul. 2017.

15. Shota Yusa, Takafumi Katayama, Wen Shi, Tian Song and Takashi Shimamoto : Fast CU Depth Decision Algorithm Using Depth-Map for 3D-HEVC, Proceedings of International Technical Conference on Circuits/Systems, Computers and Communications(ITC-CSCC2017), 473-474, Busan, Jul. 2017.

16. Takafumi Katayama, Wen Shi, Tian Song and Takashi Shimamoto : Early Depth Determination Algorithm for Enhancement Layer Intra Coding of SHVC, Proceedings of IEEE International Technical Conference (TENCON 2016), Singapore, Nov. 2016.

17. Kazuki Kuroda, Takafumi Katayama, Tian Song and Takashi Shimamoto : Adaptive Mode Selection for Low Complexity Enhancement Layer Encoding of SHVC, Proceedings of International Technical Conference on Circuits/Systems, Computers and Communications(ITC-CSCC2016), Okinawa, Jul. 2016.

18. Takafumi Katayama, Tian Song, Wen Shi, Takashi Shimamoto and Jenq-Shiou Leu : Reference Frame Selection Algorithm of HEVC Encoder for Low Power Video Device, Proceedings of 2nd International Conference on Intelligent Green Building and Smart Grid (IGBSG 2016), Praha, Jun. 2016,

19. Takafumi Katayama, Wen Shi, Tian Song, Jenq-Shiou Leu and Takashi Shimamoto : Fast CU Size Decision for Intra Coding Algorithm in SHVC, Proceedings of 2nd International Forum on Advanced Technologies (IFAT2016), Tokushima, Mar. 2016.

20. Takafumi Katayama, Wen Shi, Tian Song and Takashi Shimamoto : Low-Complexity Intra Coding Algorithm in Enhancement Layer for SHVC, Proceedings of 2016 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, Jan. 2016.

21. Takaaki Hamamoto, Takafumi Katayama, Tian Song and Takashi Shimamoto : Novel Variable Search Range Selection Algorithm for H.264/SVC, Proceedings of International Workshop on Nonlinear Circuits, Communication and Signal Processing (NCSP'11), 316-319, Tianjin, China, Mar. 2011.

22. Takafumi Katayama, Takaaki Hamamoto, Tian Song and Takashi Shimamoto : Motion Estimation Algorithm for Spatial Scalability in H.264/SVC, Proceedings of International Workshop on Nonlinear Circuits, Communication and Signal Processing (NCSP'11), Tianjin, China, Mar. 2011.

23. Takafumi Katayama, Takaaki Hamamoto, Tian Song and Takashi Shimamoto : Motion Based Low Complexity Algorithm for Spatial Scalability of H.264/SVC, Proceedings of 2010 IEEE 17th International Conference on Image Processing (ICIP2010), Hong Kong, Sep. 2010.

24. Kentaro Takei, Naoyuki Hirai, Takafumi Katayama, Tian Song and Takashi Shimamoto : Real-Time Architecture for Inter-layer Prediction of H.264/SVC,

Proceedings of Pacific-Rim Conference on Multimedia (PCM2010), Shanghai, Sep. 2010.

25. Kentaro Takei, Takafumi Katayama, Tian Song and Takashi Shimamoto : Complexity Reduction Algorithm for Enhancement Layer of H.264/SVC, The Third International Symposium on Intelligent Informatics (ISII2010), Dalian, Sep. 2010.
26. Takafumi Katayama, Takaaki Hamamoto, Tian Song and Takashi Shimamoto : Low Complexity Inter-Layer Motion Estimation Algorithm for H.264/SVC，Proceedings of World Automation Congress (WAC2010), Kobe, Sep. 2010.
27. Yoshitaka Morigami, Tian Song, Takafumi Katayama and Takashi Shimamoto : Low Complexity Algorithm for Inter-layer Residual Prediction of H.264/SVC, IEEE International Conference on Image Processing (ICIP2009), Cairo, Egypt, Nov. 2009.
28. Takaaki Hamamoto, Takafumi Katayama, Tian Song and Takashi Shimamoto : Mode Selection Method for Hierarchical B-picture of H.264/SVC, The Second International Symposium on Intelligent Informatics (ISII2009), QinHuangDao, China, Sep. 2009.
29. Takafumi Katayama, Yoshitaka Morigami, Tian Song and Takashi Shimamoto : Improvement of Motion Estimation with Modified Search Center and Search Range for H.264/SVC，The 24th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC2009), Cheju, Jul. 2009.

Acknowledgement

First of all, I would like to express my gratitude to my advisors, Associate Professor Tian Song and Professor Takashi Shimamoto, 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 would like to extend my appreciation to Professor Masaki Hashizume, Professor Yoshifumi Nishio, Associate Professor Hiroyuki Yotsuyanagi, Associate Professor Yoko Uwate, and other technical staffs in our course, for their supports and friendship.

I would like to express my sincere appreciation to all the members of SimaSong laboratory.

I am deeply grateful to Yume scholarship from Tokushima University for their financial support. I want to give my sincerely thanks to my family, my father and my mother, thanks for your always supporting in my study. Without you, it would be impossible for me to gain these achievements. I wish you all a healthy body.

Finally, I would like to express my gratitude to all people who support me.

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