Master's Thesis

Next-Generation Scalable Video Coding

By

Ruoyang Yu

Department of Electrical and Information Technology
Faculty of Engineering, LTH, Lund University
SE-221 00 Lund, Sweden

Abstract

Scalable video coding's goal is to provide universal video access in heterogeneous networks. The H.264/SVC coding standard is widely used to provide a single video bitstream whose decoded video sequences could be different in resolution, frame rate and quality. The next generation video coding standard HEVC is a successor to the H.264/AVC standard and there are plans to develop its scalability extension.

In this thesis, spatial scalability tools in H.264/SVC such as Inter Layer Intra Prediction and Inter Layer Motion Prediction were studied as well as their applicability and performance in HEVC. Similar to H.264/SVC's Base Mode for macroblocks, the Base Mode for Coding Unit in HEVC is also included in the design. In order to evaluate performance of SVC tools in HEVC, a test environment for testing scalable tools in HEVC was implemented.

The design supported two layer spatial scalability with 1:2 ratio in each resolution dimension. Results showed that the inter layer intra prediction could help achieve significant gains in intra prediction cases and also provided most of gain for inter prediction when multi-loop decoding was enabled. The inter layer motion prediction and Base Mode results indicated less gains than in case of H.264/SVC. Therefore, new tools that are suitable for HEVC need to be developed in the future.

Acknowledgments

This Thesis was done at Ericsson Research, Kista, Sweden.

First and foremost, I would like to express my gratitude to my supervisors, Andrey Norkin and Kenneth Andersson, for their invaluable guidance and suggestions. Their positive spirit and enthusiasm for video coding research stimulate me a lot.

I am grateful to everyone in the Multimedia Technologies department for provide a pleasant working environment. They make me feel welcome and make my thesis time at Ericsson Research enjoyable and unforgettable.

Many thanks also go out to Irina Bocharova for her precious feedback and criticism on my report.

Ruoyang Yu
Stockholm, May 2012.

Table of Contents

CHAPTER 1

1 Introduction

1.1 Background

The video coding technology's continuous development comes along with various video applications, such as real-time video conferencing, television broadcasting. These applications might be employed over different scenarios, different channels like wireless mobile network or wired internet, receiving devices like mobile phone or TV with different displaying screen and different processing capabilities. In order to achieve these flexible purposes, Scalable Video Coding (SVC) is considered as a possible solution. The SVC's video encoder can provide a scalable video bit-stream which can adapt to various needs of endpoints in diversity of networks.

SVC has been investigated over last two decades. The international video coding standards H.262, H.263, MPEG-4 Visual, and H.264 have already included coding tools to support scalabilities. In these coding standards, the scalable extension of H.264/AVC (Advanced Video Coding), also known as H.264/SVC, which was jointly developed by ITU-T VCEG and the ISO/IEC MPEG groups, is currently considered to be the state-of-the-art of scalable video standard. It has achieved significant improvements on reducing codec complexity while improving scalable coding efficiency.

In January 2010, the VCEG and MPEG group (called JCT-VC) jointly launched a project aiming for a new video coding standard [1]. This next generation video coding standard is called HEVC (High Efficiency Video Coding) and the goal of this standard is to reduce the bit-rate of H.264/AVC by half without compromise on video quality. This emerging standard incorporates a lot of new compression techniques and algorithms and it is still under development. The final standard is expected to be ready by January 2013 and a joint call for proposals on scalable extension of HEVC is planned to be issued in July 2012.

1.2 Thesis outline

The goal of this thesis is to study scalable video coding technologies in H.264/SVC and HEVC video coding technology, to implement known scalability video coding tools on top of HEVC and evaluate its performance.

Chapter 2 gives some theoretical background referred in this thesis, including the basic concept of HEVC and SVC.

Chapter 3 presents a performance study of inter layer prediction in H.264/SVC.

Chapter 4 introduces the key part of this thesis—implementation and evaluation of different spatial scalability tools in HEVC. Some analyses of the simulation results are also presented.

Chapter 5 is the conclusion chapter and proposes some new inter layer prediction ideas for future work.

CHAPTER 2

2 Video coding concepts and definitions

This chapter is divided into two parts. The first part describes essential concepts and definitions in the scope of HEVC [6]. The second part describes the scalable video coding concepts and scalable tools in H.264/SVC.

2.1 High Efficiency Video Coding (HEVC)

The motivation of video coding is to compress a source video sequence by removing its redundancy information. A general video codec consists of an encoder and a decoder, as shown in Figure 2-1. The encoder converts the video sequence into a compressed bitstream which makes it easy for storage or transmitting over a limited network bandwidth. The decoder decodes the compressed bitstream and reconstructs video sequence. Most of video coding standard uses lossy compression, which gives higher compression while maintaining the reconstructed video at an acceptable quality level.

Figure 2-1 Video Codec

2.1.1 HEVC codec

HEVC is a video coding industry standard. It describes a set of video coding tools and defines a format for coded video bitstream. The HEVC standard uses block-based hybrid coding to exploit both spatial and temporal redundancy information for compression. A general HEVC codec is shown below. The main components will be gone through in following sub-sections.

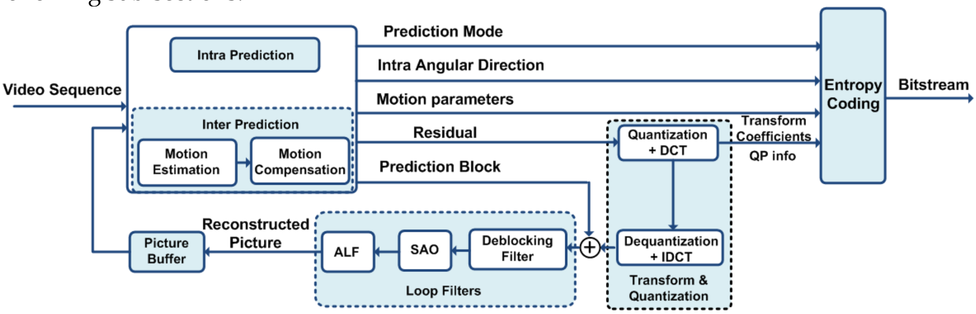


Figure 2-2 HEVC encoder structure

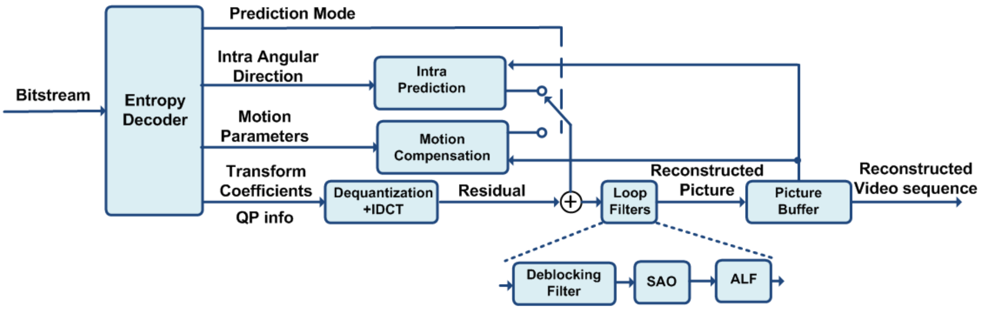


Figure 2-3 HEVC decoder structure

2.1.2 Colour space

There are several alternative ways to represent a pixel in one picture of a video sequence. Each of them is known as a colour space.

YUV colour space and its 4:2:0 sampling pattern are commonly used in video coding. A pixel is represented by one luminance component Y and two chrominance components U and V. The 4:2:0 sampling pattern means that the chroma components U and V each has half resolution of Y both in horizontal and vertical direction. This separation of luma and chroma components can improve compression efficiency since the human visual system (HVS) is more sensitive to luminance than chrominance.

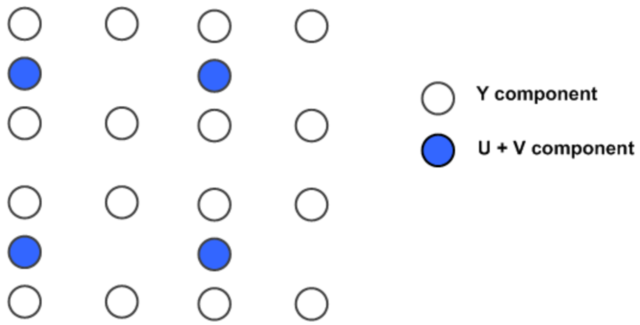


Figure 2-4 YUV colour space with 4:2:0 sampling

2.1.3 Picture partitioning

2.1.3.1 Pictures

A picture (also called as frame) is a still image within a video sequence. Each picture is encoded and there are three coded picture types: I-picture (intra coded), P-picture (inter coded) and B-picture (inter coded).

An I-picture is coded without any reference to other pictures. A P-picture is a picture coded with reference to one previously decode picture which can be either earlier in viewing order or after in viewing order. A B-picture is a picture coded with reference to one or two previously decoded pictures.

2.1.3.2 Coding Unit

Pictures are divided into a sequence of largest coding unit (LCU). The Maximum largest coding unit's size is typically 64x64. A LCU can be further split into Coding Units (CUs) (Figure 2-5). They are arranged in a quad-tree structure (Figure 2-6). LCU is the tree root and CU is a leaf.

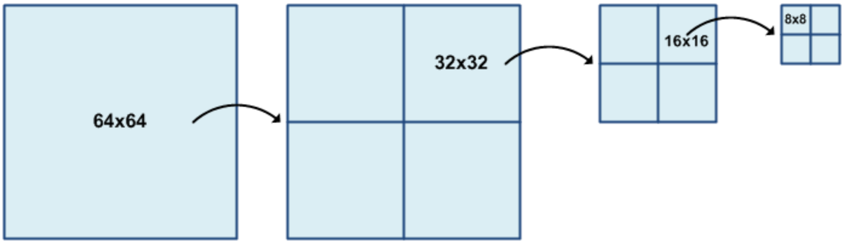


Figure 2-5 Example of CU splitting

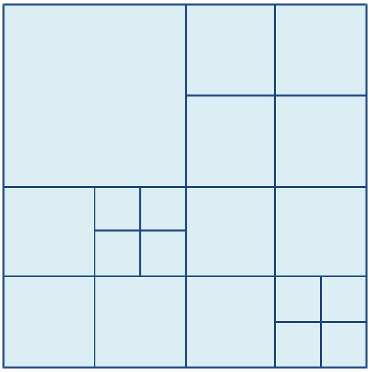


Figure 2-6 Example of LCU quad-tree structure

CU is the basic region for coding. It is square and its size can be 8x8, 16x16, 32x32 and 64x64. CU's role is similar to conventional macroblock's in H.264, but the flexible size of CU can provide better coding efficiency than fixed size macroblock since CU size could be content-adaptive. Large CU size could be suitable for a large plain region in a picture and small CU size may result in better prediction for regions that contain a lot of detail information.

2.1.3.3 Prediction Unit

A CU can be further split into Prediction Units (PUs). The Prediction Unit is the unit where prediction takes place and it carries all prediction parameters such as angular direction for intra coded PU, motion vectors, reference picture list and reference picture index for inter coded PU.

Figure 2-7shows possible PU partitions according to the prediction type for a CU size 2Nx2N. If the CU is in skip mode, the PU size should be the same as CU size which is 2Nx2N. If the CU is in intra mode, the PU may take size 2Nx2N or NxN. If the CU is in inter mode, the PU may take size 2Nx2N, Nx2N, 2NxN, NxN or using Asymmetric Motion Partition (AMP) which includes partition types like 2NxnU, 2NxnD, nLx2N, nRx2N. It is worth noticing that this asymmetric partition can bring compression efficiency since it fits irregular patterns better than using symmetric partition.

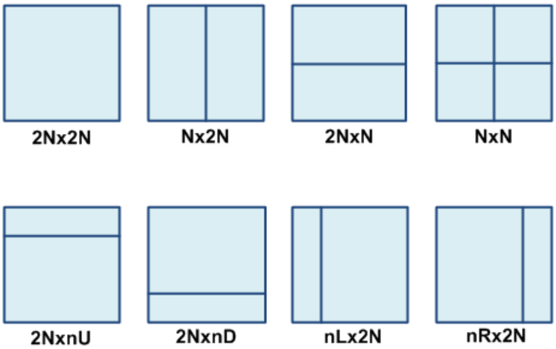


Figure 2-7 Possible PU partitions

2.1.3.4 Transform Unit

Transform Unit (TU) is the basic unit for transform and quantization. One CU may contain one or several TUs and the TUs are arranged in a quad-tree structure within CU. The maximum TU size is 32x32 and minimum TU size is 4x4. The maximum level of TU tree within a CU is 3.

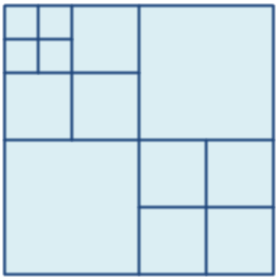


Figure 2-8 Example of TU's quad-tree structure

2.1.3.5 Slice, Tile

Slices and Tiles are a group of coding units. They can be decoded independently without information from other slices or tiles. Tiles are always rectangular, but slice shape could be flexible.

2.1.4 Predictions

2.1.4.1 Intra Prediction

Intra Prediction predicts a PU from previously coded upper and left neighbouring pixels. This prediction method is to exploit spatial redundancy information in current picture since the neighbouring pixels are highly correlated with the pixels of current block in most cases.

The intra prediction direction modes can be up to 35 including one non-directional mode (DC mode) and one planar mode. In Figure 2-9, the red small blocks represent previously coded pixels and the blue lines are the possible 33 directions. In DC mode, prediction value is the mean value of all the upper and left neighbouring pixels. In planar mode, prediction value is formed by a linear combination of upper and left pixels.

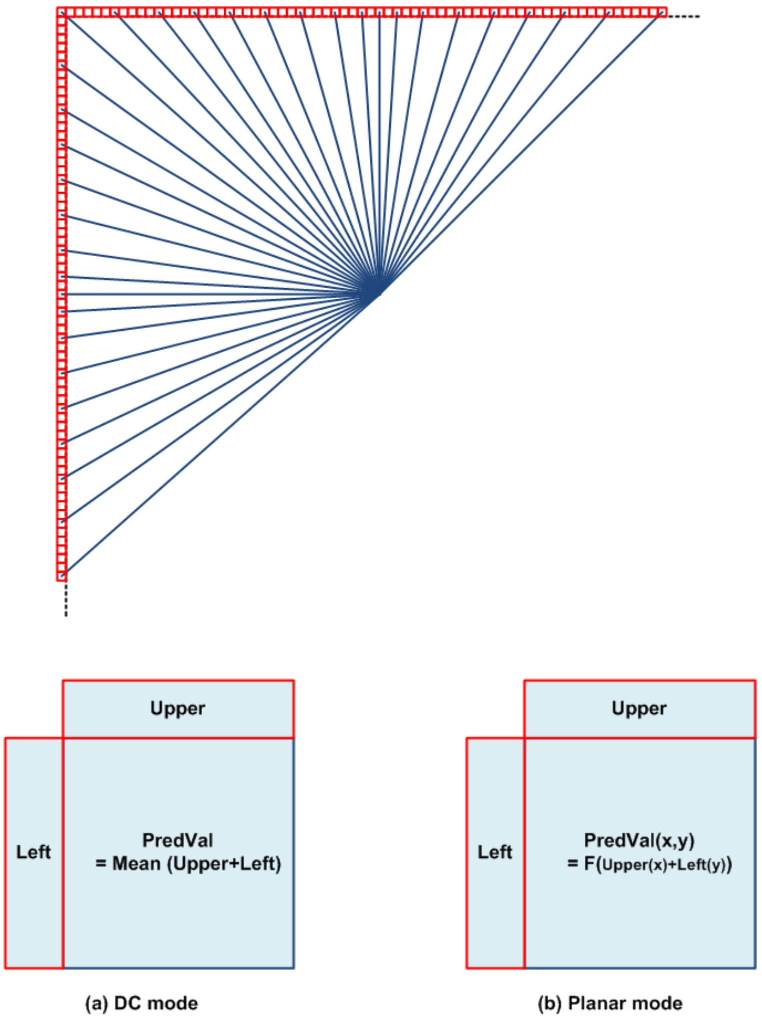


Figure 2-9 Directions of intra prediction

2.1.4.2 Inter Prediction

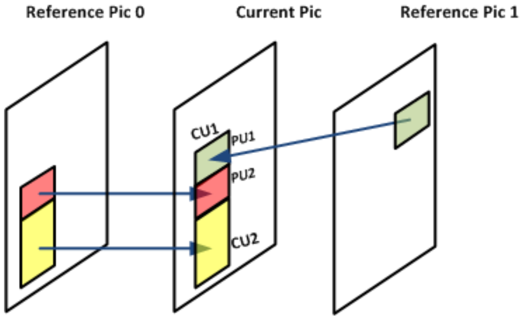


Figure 2-10 Example of Inter prediction using multiple reference pictures

Inter prediction predicts a PU from its reference pictures which have been previously decoded. This prediction method is to exploit temporal redundancy information between current picture and its reference pictures. Motion Estimation (ME) is invoked to find a region in reference pictures that has the same size of the PU and could closely match the current PU. The position of the best-match region is represented by a Motion Vector. Motion Compensation (MC) is to subtract the best-match region from the current PU and produce a residual block. As shown in Figure 2-10, CU1 is split into two PUs, PU1 and PU2. PU1 is predicted from a region in reference pic 0 while PU2 is predicted from a region in reference pic 1, the blue arrows represent motion vectors.

There are two prediction modes in inter prediction, skip mode and inter mode.

If a CU is coded with skip mode, the CU has only one PU and its motion parameters, e.g. motion vectors, reference picture list (reference picture list 0 or reference picture list 1) and reference picture index, are inferred from one of the PU's spatial and temporal neighbouring inter-coded PUs (called as motion merge). As an example, in Figure 2-10, CU2's motion vector could be derived from CU1's PU2. The skipped CU just copies the corresponding block from the reference picture and no residual data is transmitted. The HEVC encoder chooses the best motion parameters from a candidate list. The candidate list is formed by four spatial neighbouring PUs and one temporal neighbouring PU. Four spatial merge candidates are selected among candidates located in positions as shown in Figure 2-11. The order of derivation is A1 → B1 → B0 → A0 → (B2). The position B2 is considered when any of the positions A1, B1, B0 and A0 is not inter-coded or not available, i.e. outside the current picture or slices. The temporal candidate is chosen between two candidates located in positions as Figure 2-11. The order of derivation is D0 → (C0). C0 is used when position D0 is inter-coded, not available, or outside the current LCU.

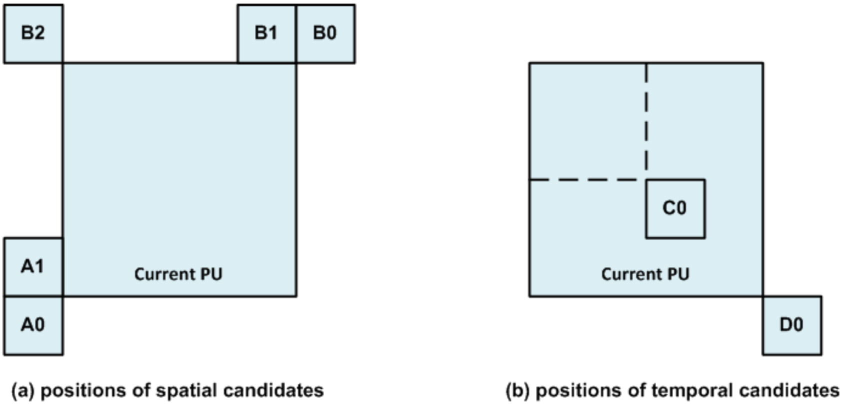


Figure 2-11 Candidate positions for motion merge

If a CU is coded with inter mode, its PUs are predicted by motion compensation and its motion parameters can be derived implicitly using motion merge method or transmitted explicitly. Motion vectors of inter-coded CU could be predicted by motion vector prediction method and the motion vector difference is transmitted. The encoder chooses the best motion vector predictor among a candidate list formed by two spatial candidates (one from top and one from left) and one temporal candidate. The candidate positions are the same as in motion merge. The derivation order of the spatial candidate from top is B0 → B1 → B2 and the order for candidate from left is A0 → A1. The derivation of the temporal candidate is the same as in motion merge. Unlike the skipped CU, the residual information of inter-coded CU is transmitted.

2.1.5 Transform and Quantization

After intra and inter prediction, the pixels of residual blocks are transformed and the resulting coefficients are scalar quantized. The motivation of transform operation is that the residual pixels tend to be correlated and the energy is likely to be evenly distributed in spatial domain, which makes it hard to compress. Transform operation decorrelates these data and localizes the majority of energy into a small portion of coefficients. The coefficients with small energy could be discarded without significantly affecting quality since they contain little information.

Transform in HEVC are mainly based on integer Discrete Cosine Transform (DCT). The 2-dimensional DCT is determined as following equations, $\boldsymbol{X}$ is the input NxN residual block, $\boldsymbol{A}$ is transform matrix and $\boldsymbol{Y}$ is transform coefficients block. The significant DCT coefficients are typically at low frequency positions and are clustered around top-left of the coefficient block. Coefficients that correspond to high-frequency are small and are near zero.

$$\boldsymbol{Y} = \boldsymbol{AXA^T}, \boldsymbol{A}(\boldsymbol{i}, \boldsymbol{j}) = \boldsymbol{C}(\boldsymbol{i})\sqrt{\frac{2}{\boldsymbol{N}}}\cos\left(\frac{(2\boldsymbol{j}+1)\boldsymbol{i}\pi}{2\boldsymbol{N}}\right), \boldsymbol{i}, \boldsymbol{j} = 0,1,\ldots \boldsymbol{N}-1 \quad (2.1)$$

$$\boldsymbol{C}(\boldsymbol{k}) = \begin{cases} 1/\sqrt{2}, & \boldsymbol{k} = 0 \\ 1, & \boldsymbol{k} > 0 \end{cases} \quad (2.2)$$

A disadvantage of using transform matrix $\boldsymbol{A}$ is that it contains irrational numbers. It would be complexity-saving to have a transform matrix with integer entries. The integral DCT is to scale each row of $\boldsymbol{A}$ and rounding to the nearest integer.

$$\boldsymbol{A_{integer}} = \boldsymbol{round}(\alpha \cdot \boldsymbol{A}) \quad (2.3)$$

For example, a 4x4 transform matrix is:

$$\boldsymbol{A} = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.653 & 0.271 & 0.271 & -0.653 \\ 0.5 & -0.5 & -0.5 & 0.5 \\ 0.271 & -0.653 & -0.653 & 0.271 \end{bmatrix}$$

A multiply by 128 and round gives a 4x4 transform matrix of HEVC:

$$\boldsymbol{A_{integer}} = \begin{bmatrix} 64 & 64 & 64 & 64 \\ 83 & 36 & -36 & -83 \\ 64 & -64 & -64 & 64 \\ 36 & -83 & 83 & -36 \end{bmatrix}$$

Quantization process is invoked after applying transform to remove non-informative transform coefficients with small values. The quantization uses scalar quantizer and this is a lossy step. The quantization step size depends on Quantization Parameter (QP). High QP value corresponds to a small dynamic range of quantized values and results in less bit cost, whereas the distortion is high. Low QP value corresponds to a larger range of quantized values and results in higher bit cost, whereas the distortion is low since the reconstructed value could match original value better than using higher QP. In HEVC, QP value could take a range from 0 to 51. The quantization step size increases twice with every increment of 6 in QP.

$$\boldsymbol{F_{Quantized}} = \boldsymbol{Round}\left(\frac{\boldsymbol{F_{Original}}}{\boldsymbol{QP}}\right) \quad (2.4)$$

$$\boldsymbol{F_{Reconstruted}} = \boldsymbol{QP} \cdot \boldsymbol{F_{Quantized}} \quad (2.5)$$

2.1.6 Loop Filtering

Loop Filters are Deblocking Filter, Sample Adaptive Offset (SAO) and Adaptive Loop Filter (ALF). These loop filters are mainly used for reducing the artefacts and distortion caused by predictions, block-based transforms and quantization. The deblocking filter considers the block boundary, SAO considers non-linear filtering using edge and pixel categorization or pixel value instead of intensity and ALF considers Wiener filtering.

The deblocking filter's function is similar to that in H.264 to filter over the block edge pixels in order to reduce blocking distortion. Figure 2-12 shows an example: the deblocking filter preserves sharp object edges in picture (a); in picture (b), some edges of objects are blurred.

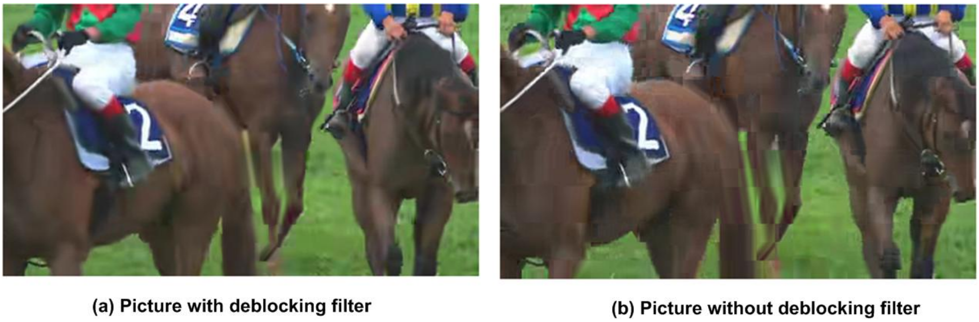


Figure 2-12 Racehorse coded with QP=37

After deblocking filter, SAO is applied. The SAO process recursively splits the current picture into different sub-regions where each sub-region is determined to either modify pixel intensities in specific pixel intensity ranges or modify pixel value according to edge properties. SAO reduces the distortion by adding a transmitted offset to respective pixel intensity of each category.

ALF is the final stage of loop filtering to further reduce the distortion between reconstructed picture and original picture. If ALF are applied, filter coefficients are explicitly encoded.

2.1.7 Entropy Coding

Entropy coding is used for further compressing all the information concerning the prediction parameters, transform coefficients etc. The coding method is CABAC (Context Adaptive Binary Arithmetic Coding).

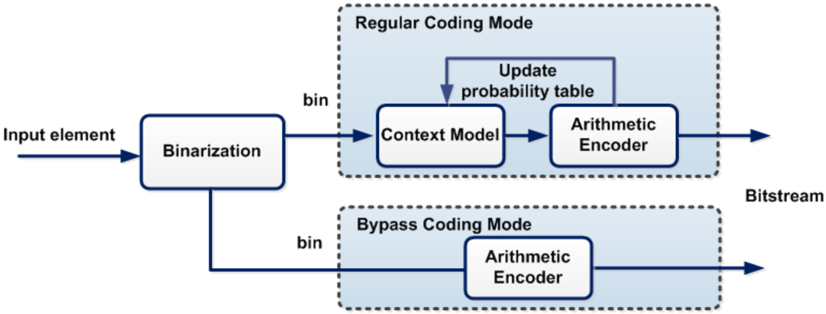


Figure 2-13 CABAC coding structure

In CABAC encoder, the inputs to arithmetic coding are binarized value, e.g. only two sub-ranges corresponding to the probability of 1 or 0. Each binarized input value is called a bin. As an example, Table 2-1 lists the probabilities of a CU's skip_flag being 0 or 1, assuming a sequence of bins is (1, 1, 1, 1).

Table 2-1. An example of SKIP flag's probability table

SKIP_flag	Probability(P)	-log2(P)
0	0.2	2.32
1	0.8	0.32

Figure 2-14 shows the binary arithmetic coding process. The binary arithmetic coding starts with the probability model of bin being 0 or 1, each bin value is assigned to a sub-region in region 0 to 1 according to corresponding probability. The region (0, 1) will be progressively divided into new sub-region according to incoming symbols. The first symbol 1 selects the sub-region (0.2, 1). The next symbol 1 selects the sub-region (0.36, 1) as new sub-region based on previous sub-region, and so on. The final sub-region is (0.5904, 1) and any number falling within this range could represent the entire data sequence, for example 0.75. The number 0.75 could be represented as a fixed-point fractional number using 3 bit. The arithmetic coding's efficiency depends on the accuracy of probability model. An inaccurate probability model could introduce large bit cost. For example, the encoder has a wrong probability table with the probability of 0 to 0.8 and probability of 1 to 0.2. Then, the final sub-region would be (0.9984, 1) and the bit cost reaches 11 bits!

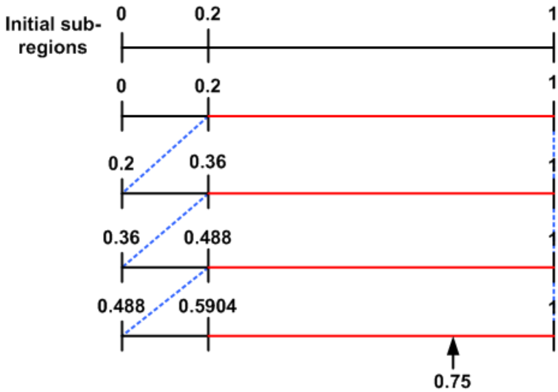


Figure 2-14 Arithmetic coding example

In order to get an accurate probability model for coding each bin, context adaptive coding is added on top of binary arithmetic coding. The motivation of context adaptive coding is to have an "up-to-date" probability table which is coherent with local statistics. There are two coding modes for coding a bin: regular coding mode and bypass coding mode [7]. In regular coding mode, a context model should be chosen before encoding a bin. The context model stores probabilities information of each bin being 1 or 0. The choice of context models depends on the recently coded data symbols. Then, the arithmetic coder encodes each bin according to the selected context model and updates the selected context model's probability information. For example, three context models for skip_flag is shown in Table 2-2 and Figure 2-15. The choice of context model depends on two previously coded neighbouring block A to the left and B from above: $\boldsymbol{I} = \boldsymbol{skip\_flag}(\boldsymbol{A}) + \boldsymbol{skip\_flag}(\boldsymbol{B})$. When $\boldsymbol{I} = 0$, there is high probability that current CU has $\boldsymbol{skip\_flag} = 0$ and model 0 with a probability table of $\boldsymbol{skip\_flag}$ being 0 larger than being 1 would be chosen.

Table 2-2. An example of context models for skip_flag

I	Context model
0	Model 0
1	Model 1
2	Model 2

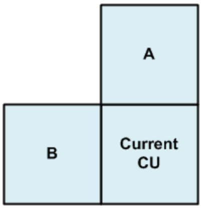


Figure 2-15 Example of a context template for skip flag

In bypass coding mode, bins are directly coded using arithmetic coding without context modelling. The initial probability of bin being 0 or 1 is equal probable, e.g. 0.5.

2.1.8 Rate-Distortion Optimization

Encoder chooses the best encoding parameters by finding modes and predictions which can give minimum R-D cost. The R-D cost function is calculated as $\boldsymbol{J} = \boldsymbol{D} + \lambda \cdot \boldsymbol{R}$, where D is distortion, R is the bits cost and λ is a Lagrangian constant.

There are three measurements for distortion, Sum of Square Error (SSE), Sum of Absolute Difference (SAD) and Hadamard Transformed SAD (SATD).

$$SSE = \sum_{i,j} (P_{prediction}(i,j) - P_{original}(i,j))^2 \tag{2.6}$$

$$SAD = \sum_{i,j} \left| (P_{prediction}(i,j) - P_{original}(i,j)) \right| \tag{2.7}$$

$$SATD = \left(\sum_{i,j} \left| T(P_{prediction}(i,j) - P_{original}(i,j)) \right| \right) / 2 \tag{2.8}$$

In intra/ inter mode decision, the R-D cost J_{mode} is calculated as in equation 2.5, where the distortion considers both distortion of luma and chroma, and w_{chroma} is a weighting factor.

$$J_{mode} = (SSE_{luma} + w_{chroma} \cdot SSE_{chroma}) + \lambda_{mode} \cdot R_{mode} \tag{2.9}$$

2.2 *Scalable Video Coding (H.264/SVC)*

2.2.1 H.264/SVC concepts

In scalable video coding, there is a layer structure: one base layer (BL) and one or more enhancement layers (EL). The base layer refers to a basic resolution/quality/frame rate video sequence and enhancement layer refers to video sequences that have higher resolution/quality/frame rate compared to base layer. The base layer is encoded independently while enhancement layers are encoded with dependency on the base layer's information. These coded layers information are merged into one single scalable video stream. A scalable video stream refers to those parts of the video stream could be removed and the remaining substream is still decodable. The substream that corresponds to base layer's information provides a reconstruction video sequence with basic resolution/quality/frame rate. The substream with enhancement layers information could be decoded and added to the base layer to provide higher resolution (named as spatial scalability, as shown in Figure 2-16), frame rate (named as temporal scalability, as shown in Figure 2-17) or quality (named as quality scalability, as shown in Figure 2-18). Therefore, the scalable video stream gives potential users in a heterogeneous network freedom to choose which substream to decode and reconstruct according to individual preferences. As already indicated above, there are three types of scalability: temporal scalability, quality scalability and spatial scalability, which will be introduced as following.

Figure 2-16 An example of two temporal layers

Figure 2-17 An example of two spatial layers

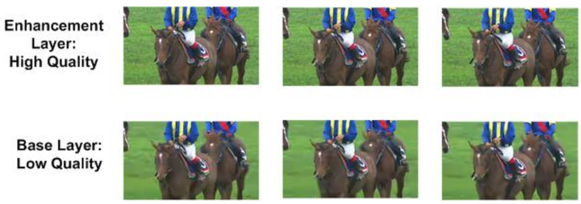


Figure 2-18 An example of two quality layers

2.2.2 Temporal Scalability in H.264/SVC

Temporal scalability allows a single bitstream to support different frame rates. It is supported with hierarchical-B prediction structure in H.264/AVC. Figure 2-19 shows a typical hierarchical-B prediction structure with four temporal layers L0, L1, L2 and L3. L0 is the base layer with lowest frame rate. L1, L2 and L3 are enhancement layers. Pictures at enhancement layers are encoded as B-pictures and their reference pictures are from lower temporal layer. As an example, Layer 2's reference pictures are from Layer 0 and Layer 1. In this hierarchical prediction structure, the pictures at base layer are coded using lowest quantization parameter to provide highest quality, since they serve as reference pictures for all temporal layers. The quantization parameter increases with increase of temporal level, e.g. pictures at higher layer are coded with higher quantization parameter since the possibility of these pictures serving as reference pictures is lower.

A target decoder could decode and reconstruct video sequence with different frame rate by choosing different number of layers to decode. One instance is that base layer L0 with frame rate of 7.5 fps, after decoding enhancement layer L1 and added with L0, the reconstructed video sequence's frame rate could achieve 15 fps.

The temporal scalability has already been included in HEVC using hierarchical-B prediction structure.

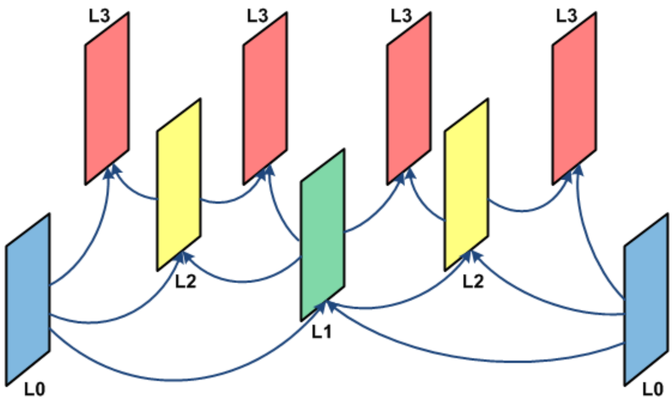


Figure 2-19 Hierarchical-B prediction structure

2.2.3 Spatial Scalability in H.264/SVC

The H.264/SVC encoder structure with two spatial layers is showed in Figure 2-20. The enhancement layer is high resolution video sequences and base layer is low resolution video sequences. The SVC encoder incorporates Inter-Layer Prediction tools to enable the usage of base layer's information as much as possible. There are three inter layer prediction tools: Inter layer Intra Prediction (ILIP), Inter Layer Motion Prediction (ILMP) and Inter Layer Residual Prediction (ILRP) [8]. These inter layer prediction tools' motivation is to find the redundancy information between EL and BL since the two layers' content properties are similar. The encoder could choose coding a macroblock using these inter layer prediction tools or using original single layer coding tools.

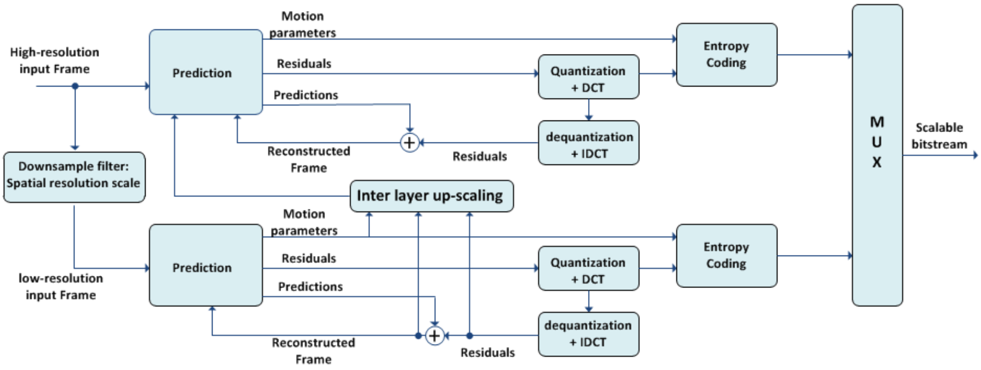


Figure 2-20 SVC encoder structure

Besides these inter layer prediction tools, H.264/SVC introduced another prediction mode for enhancement layer's macroblock: Base Mode. If a macroblock is in base mode, only its residual signal is encoded and other information such as motion parameters and prediction modes are inferred from base layer using inter layer prediction tools.

If a macroblock is in base mode, e.g. its base mode flag is set to 1, first the encoder should find its co-located BL's macroblock according to the coordinates. If its co-located macroblock is intra-coded, then the EL's macroblock is intra-coded and uses the up-sampled BL's reconstructed picture for prediction, i.e. Inter Layer Intra Prediction. If the co-located BL's macroblock is inter-coded, then the EL's macroblock is inter-coded and its sub-macroblock partitions, motion vectors and reference index are inferred from base layer using Inter Layer Motion Prediction.

The usage of up-scaled reconstructed picture for prediction is limited for those parts of the BL which are intra-coded. This restriction only requires one motion compensation loop for EL, which is called single-loop decoding. Multi-loop decoding refers to that the EL could use the whole up-scaled reconstructed picture without limitation. In this case, BL need one more motion compensation loop to reconstruct macroblocks that are inter-coded. Single loop decoding could reduce the complexity but introduce some penalties on performance.

Besides the base mode flag, there is a residual prediction flag added to macroblock syntax. If the EL's macroblock's residual prediction flag is set to 1, the residual signal of the collocated block is upsampled and used as a prediction for the macroblock's residual signal.

The spatial scalability, which is not included in current HEVC, is mainly focused on in this thesis.

2.2.4 Quality Scalability in H.264/SVC

Quality scalability aims at providing a single bitstream to support different quality level, e.g. a base layer provides with basic video quality and enhancement layers with higher video quality. A general quality scalability mode is called Coarse Grain Scalability (CGS). The CGS could be viewed as a special case of spatial scalability with layers that have same resolution. The same inter-layer prediction tools as in spatial scalability but without up-sampling operation are used. There are other two quality scalability modes as variations of CGS: Medium Grain Scalability (MGS) and Fine Grain Scalability (FGS). [8] gives more detailed information on MGS and FGS.

2.3 *Simulation and evaluation specifications*

2.3.1 HEVC Reference Software

The HEVC Reference Software is developed by members of JCT-VC and is named HM (HEVC test model). HM version 5.0 has been used in this thesis. The HM5.0 was defined by decisions taken at the 7th meeting of JCT-VC and the working draft of HEVC can be found in [2]. The Software is written in C++ and its source code is available for download at [3].

2.3.2 Encoder Configurations

The HM encoder has six kinds of configurations, All-Intra, Low-Delay and Random-Access each with high efficiency coding and low complexity coding configuration. In this thesis, the tests were done with high efficiency configuration.

High Efficiency coding

The high efficiency coding aims for high compression performance but introduces high computational complexity. All loop filters are enabled.

All-Intra configuration

In All-Intra configuration, each picture is encoded as I-picture.

Low-Delay configuration

In Low-Delay configuration, the first picture is encoded as I-picture and the rest pictures are encoded as P-pictures or B-pictures with reference restricted to previous decoded pictures earlier in viewing order.

Random-Access configuration

Random-Access configuration uses hierarchical B coding structure. I-pictures are inserted periodically and pictures in between are encoded as B-pictures.

2.3.3 Objective Metrics

Objective measurements are used for results evaluation. Two main measurement methods are Peak Signal to Noise Ratio (PSNR) and BD-Rate [4].

PSNR is calculated according to the equation 1.1 and 1.2. The Mean Squared Error (MSE) is measured between the anchor video picture and test video picture pixel values. The variable H and W represent height and width of pictures in a video sequences. $\boldsymbol{P_{anchor}}$ and $\boldsymbol{P_{test}}$ represent a single pixel value for an anchor picture and a test picture respectively. $(2^{\boldsymbol{n}}-1)$ is the highest-possible pixel value in a picture where $\boldsymbol{n}$ represents the number of bits per pixel.

$$\boldsymbol{MSE} = \frac{1}{\boldsymbol{HW}} \sum_{\boldsymbol{i}=0}^{\boldsymbol{H}-1} \sum_{\boldsymbol{j}=0}^{\boldsymbol{W}-1} \left(\boldsymbol{P_{anchor}}(\boldsymbol{i},\boldsymbol{j}) - \boldsymbol{P_{test}}(\boldsymbol{i},\boldsymbol{j})\right)^2 \quad (1.1)$$

$$\boldsymbol{PSNR} = 10\log_{10} \frac{\left(2^{\boldsymbol{n}}-1\right)^2}{\boldsymbol{MSE}} \quad (1.2)$$

BD-Rate compares PSNR-Rate curves of test and anchor. It measures the bit rate gap between test and anchor while keeping the PSNR at the same level. Negative BD-Rate means the test performs better than anchor since it costs less bit to achieve same quality level. Positive BD-Rate means the test performs worse than anchor.

In Scalable Video Coding with spatial scalability performance evaluation, coding cost is the BD-Rate between single-layer coding and scalable coding while coding gain is BD-Rate between simulcast coding and scalable coding, as shown in Figure 2-21. For scalable coding and simulcast coding, the enhancement layer's PSNR is used and the bit rates are the total bit rates of enhancement layer and base layer. For single layer coding, enhancement layer's PSNR and bit rate are used.

When calculating the coding cost, single-layer coding is the anchor and the scalable coding's operating points were estimated by a log-linear fit to match single-layer's PSNR. When calculating the coding gain, simulcast coding is the anchor and the scalable coding's operating points were estimated by a log-linear fit to match simulcast coding's PSNR.

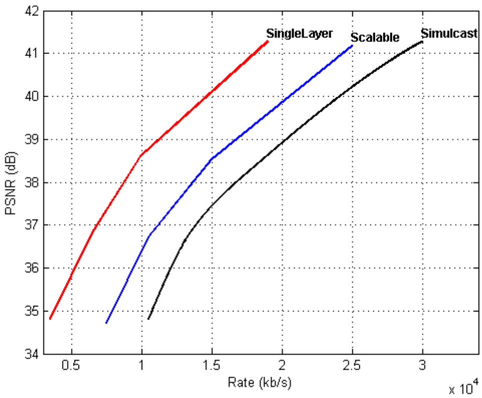


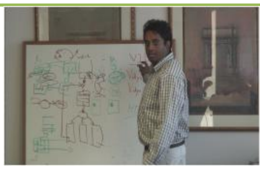
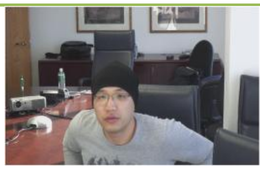
Figure 2-21 Example of R-D curves for svc performance evaluation

2.3.4 Video Test Sequences

The video test sets are provided by JCT-VC [5]. The test sets contains sequences that have different resolution, contents, frame rate and frame number.

Table 2-3. Screenshots of video test sequences

<table>
<tr><td colspan="3">Class A</td></tr>
<tr><td>
PeopleOnStreet_2560x1600_30fps</td><td>
Traffic_2560x1600_30fps</td><td></td></tr>
<tr><td colspan="3">Class B</td></tr>
<tr><td>
BasketballDrive_1920x1080_50fps</td><td>
BQTerrace_1920x1080_60fps</td><td>
Cactus_1920x1080_50fps</td></tr>
<tr><td>
Kimono1_1920x1080_24fps</td><td>
ParkScene_1920x1080_24fps</td><td></td></tr>
<tr><td colspan="3">Class C</td></tr>
<tr><td>
BasketballDrill_832x480_50fps</td><td>
BQMall_832x480_60fps</td><td>
PartyScene_832x480_50fps</td></tr>
<tr><td>
RaceHorses_832x480_30fps</td><td></td><td></td></tr>
<tr><td colspan="3">Class D</td></tr>
<tr><td>
BasketballPass_416x240_50fps</td><td>
BlowingBubbles_416x240_50fps</td><td>
BQSquare_416x240_60fps</td></tr>
<tr><td>
RaceHorses_416x240_30fps</td><td></td><td></td></tr>
</table>

Class E		
		
Vidyo1_1280x720_60fps	Vidyo3_1280x720_60fps	Vidyo4_1280_720

CHAPTER 3

3 Performance Study of Inter Layer Prediction in H.264/SVC

In this chapter, H.264/SVC's inter layer prediction tools performance in spatial scalability is studied. The H.264/SVC reference software is JSVM version 9.19.14. The test sequences are Class C and Class D and are encoded with Scalable High Profile. Since H.264/SVC incorporates inter layer mode prediction together with inter layer intra prediction (ILIP) and inter layer motion prediction (ILMP), All Intra and Low Delay P coding configuration were chosen in order to evaluate ILIP's and ILMP's performance separately from inter layer mode prediction.

3.1 Different QP configuration between layers

The first test is to investigate how choice of QP difference (dQP) between two layers affects performance, e.g. $dQP = QP_{enhance} - QP_{base}$. QPs for enhancement layer is [22, 27, 32, 37] in order to cover a PSNR region of 30dB~40dB. The base layer's QP is set to 0, 2, 4 below the enhancement layer's.

Results and Conclusions

Testing was done in H.264/SVC reference machine using All Intra high efficiency configuration and Low Delay_P high efficiency configuration. The result numbers are BD-Rate in %.

Table 3-1 shows that the coding gain and cost both increases along with the decrease in base layer's QP (increase in dQP). For all intra case with ILIP enabled, the increase in coding gain is higher than increase in cost. But for low delay case with ILMP and ILRP enabled, the increase in coding cost is much higher than the increase in coding gain.

Lower QP for base layer leads to a higher base layer quality. The increasing in base layer quality increases ILIP's efficiency since this prediction method depends on the quality of upscaled base layer pixel values largely. An upscaled reconstructed base layer with higher quality could provide better approximation of enhancement layer and results in residuals with less information. But for ILMP and ILRP, lower QP for base layer has less influence on their prediction efficiency since these two prediction are based on the upscaled of motion vectors and residuals respectively.

Table 3-1 Cost and Gains for different dQP

Cost: Single Layer vs. Scalable

Sequence	Intra High Profile (**dQP = 0**)			Intra High Profile (**dQP = 2**)			Intra High Profile (**dQP = 4**)		
	Y BD-Rate	U BD-Rate	V BD-Rate	Y BD-Rate	U BD-Rate	V BD-Rate	Y BD-Rate	U BD-Rate	V BD-Rate
Class C	13.3	13.6	14.3	15.0	15.4	15.9	17.1	17.5	17.8
Class D	13.9	14.3	15.6	16.0	16.4	17.5	18.6	18.9	19.8
Average	13.6	14.0	14.9	15.5	15.9	17.0	17.8	18.2	18.8

Gain: Simulcast vs. Scalable

Sequence	Intra High Profile (**dQP = 0**)			Intra High Profile (**dQP = 2**)			Intra High Profile (**dQP = 4**)		
	Y BD-Rate	U BD-Rate	V BD-Rate	Y BD-Rate	U BD-Rate	V BD-Rate	Y BD-Rate	U BD-Rate	V BD-Rate
Class C	-13.0	-12.7	-12.1	-16.2	-15.8	-15.4	-19.2	-18.6	-18.4
Class D	-12.1	-11.7	-10.8	-14.8	-14.3	-13.6	-17.1	-16.6	-16.1
Average	-12.5	-12.2	-11.5	-15.5	-15.1	-14.5	-18.1	-17.6	-17.2

Cost: Single Layer vs. Scalable

Sequence	Low Delay_P High Profile (**dQP = 0**) **ILMP + ILRP**			Low Delay_P High Profile (**dQP = 2**) **ILMP + ILRP**			Low Delay_P High Profile (**dQP = 4**) **ILMP + ILRP**		
	Y BD-Rate	U BD-Rate	V BD-Rate	Y BD-Rate	U BD-Rate	V BD-Rate	Y BD-Rate	U BD-Rate	V BD-Rate
Class C	22.0	20.9	21.1	27.8	26.1	26.4	34.9	32.8	33.0
Class D	22.6	21.8	22.0	29.1	27.5	28.0	36.7	34.6	35.1
Average	22.3	21.4	21.6	28.5	26.8	27.2	35.8	33.7	34.0

Gain: Simulcast vs. Scalable

Sequence	Low Delay_P High Profile (dQP = 0) **ILMP + ILRP**			Low Delay_P High Profile (dQP = 2) **ILMP + ILRP**			Low Delay_P High Profile (dQP = 4) **ILMP + ILRP**		
	Y BD-Rate	U BD-Rate	V BD-Rate	Y BD-Rate	U BD-Rate	V BD-Rate	Y BD-Rate	U BD-Rate	V BD-Rate
Class C	-3.7	-4.2	-4.2	-5.6	-6.4	-6.2	-7.5	-8.3	-8.3
Class D	-2.3	-2.9	-2.7	-3.4	-4.4	-4.1	-4.8	-5.7	-5.5
Average	-3.0	-3.6	-3.4	-4.5	-5.4	-5.2	-6.2	-7.0	-6.9

3.2 ILMP and ILRP performance

In the second test, ILMP, ILRP and combined ILIP with ILMP performance are evaluated.

Results and Conclusions

Testing was done with the reference machine using Low Delay P high efficiency configuration. QPs for enhancement layer are the same as in 3.1 and dQP was set to 2.

Table 3-2 lists coding gains and costs for different sequences. The results show that ILRP performs better than ILMP since the residual information occupies a large portion of bits in a coded bitstream. ILMP performs better for video sequences with objects moving fast or having complex motions. It is proven in cases like RaceHorse, BasketballDrill and BasketballPass, as the motion information gets higher. When ILMP and ILRP were employed together, the gain over simulcast is better than just adding the ILMP performance and ILRP performance up. The reason is when a macroblock uses a collocated base layer macroblock's motion vectors, the current macroblock's residual would be more correlated with collocated base layer macroblock's residual.

Table 3-2 Gains for different sequences in low delay P case

Gain: Simulcast vs. Scalable

Sequence	Low Delay High Profile (dQP=2) **ILMP enabled**			Low Delay High Profile (dQP = 2) **ILRP enabled**			Low Delay High Profile (dQP = 2) **ILMP + ILRP**		
	Y BD-Rate	U BD-Rate	V BD-Rate	Y BD-Rate	U BD-Rate	V BD-Rate	Y BD-Rate	U BD-Rate	V BD-Rate
BQMall_832x480_60	-1.3	-1.4	-1.3	-2.9	-3.3	-3.7	-4.6	-5.2	-5.0
PartyScene_832x480_50	-0.5	-0.7	-0.9	-2.3	-2.3	-2.2	-3.0	-3.3	-3.3
RaceHorses_832x480_30	-1.8	-1.7	-2.0	-3.6	-3.9	-4.1	-5.9	-6.2	-6.4
BasketballDrill_832x480_50	-2.0	-2.3	-2.2	-5.9	-7.5	-7.1	-8.8	-10.6	-10.2
BQSquare_416x240_60	-0.2	-0.9	-0.5	0.3	-0.7	-1.1	0.0	-1.3	-1.3
RaceHorses_416x240_30	-1.8	-1.9	-2.3	-2.9	-3.0	-3.3	-5.2	-5.5	-5.9
BasketballPass_416x240_50	-1.4	-1.8	-1.8	-4.2	-5.3	-4.1	-6.2	-7.7	-6.2
BlowingBubbles_416x240_50	-0.4	-0.6	-0.6	-1.9	-2.6	-2.4	-2.4	-3.3	-3.0
Average	-1.2	-1.4	-1.5	-2.9	-3.6	-3.5	-4.5	-5.4	-5.2

3.3 ILIP performance

In the third test, ILIP performance is studied.

Results and Conclusions

Testing was done in the reference machine using All Intra high efficiency configuration. QP settings are the same as in 3.2.

The bit rate saving over simulcast is significant about 15.5% while the cost is kept at a reasonable level. The results are as expected since in intra case, the enhancement layer could get fully use of well-reconstructed base layer, the prediction efficiency would be large.

Table 3-3 Gains for different sequences in all intra case

	Gain: Simulcast vs. Scalable			Cost: Single Layer vs. Scalable		
Sequence	All Intra High Profile (dQP = 2)			All Intra High Profile(dQP = 2)		
	Y BD-Rate	U BD-Rate	V BD-Rate	Y BD-Rate	U BD-Rate	V BD-Rate
BQMall_832x480_60	-17.2	-16.3	-16.4	17.1	18.1	18.0
PartyScene_832x480_50	-10.9	-10.9	-10.8	16.2	16.1	16.2
RaceHorses_832x480_30	-19.2	-19.1	-18.1	10.9	10.8	12.2
BasketballDrill_832x480_50	-17.5	-16.8	-16.3	15.7	16.6	17.2
BQSquare_416x240_60	-7.1	-8.6	-7.4	21.3	18.8	20.7
RaceHorses_416x240_30	-19.8	-18.1	-17.6	13.1	15.1	16.1
BasketballPass_416x240_50	-18.6	-17.0	-16.3	15.6	17.7	18.6
BlowingBubbles_416x240_50	-13.9	-13.6	-13.2	14.0	14.1	14.6
Average	-15.5	-15.1	-14.5	15.5	15.9	16.7

CHAPTER 4

4 Implementation of spatial scalable tools in HEVC

This chapter focus on introducing implemented spatial scalable tools on top of HEVC. A test-bed with scalable encoder and decoder was implemented based on the HEVC reference software. It was designed for performance tests on scalability tools rather than a real SVC codec. The codec structure contains two spatial layers, one base layer with basic resolution and one enhancement layer with high resolution. The resolution ratio between two layers is 1:2 in vertical and horizontal dimensions. The enhancement layer's input video sequences are down-sampled using the dyadic down-sampler in JSVM to create base layer video sequences. Inter layer Intra prediction, Inter Layer Motion Prediction and Base Mode similar to H.264/SVC were implemented on top of the test-bed.

4.1 SVC test-bed introduction

The motivation of implementing SVC test-bed is to have all information from the base layer ready to use when encoding and decoding the enhancement layer so that different inter layer prediction methods could be implemented on top of this test-bed and evaluated further. The SVC test-bed's encoder consists of an HEVC encoder for enhancement layer and a HEVC decoder for base layer, as shown in Figure 4-1. The inputs are enhancement layer's video sequence and base layer's bitstream which are coded off-line using normal HM5.0 encoder. In the test-bed, before encoding a picture of enhancement layer's video sequence, the base layer decoder decodes the corresponding picture that has the same picture position from the input bitstream. Afterwards, the enhancement layer encoder could get access to the base layer's decoded information such as prediction mode, partition size, motion parameters, residues and reconstructed picture. The enhancement layer's encoder could choose to encode using normal HEVC prediction or using information from base layer to exploit redundancy information between two layers.

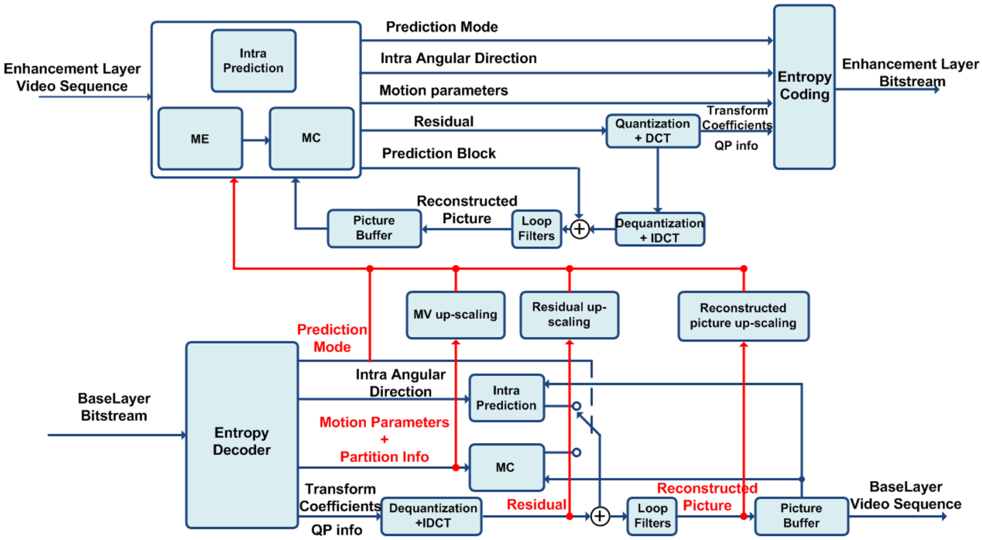


Figure 4-1 SVC test-bed encoder structure

The SVC test-bed's decoder structure is shown in Figure 4-2. An HEVC decoder for base layer is integrated into HEVC decoder for enhancement layer. The inputs to the test-bed's decoder are two layers' bitstream. Before decoding an enhancement layer's picture, the test-bed decoder decodes the base layer's picture at the same picture position. Then, the enhancement layer's decoder could use the decoded base layer's information to decode if inter layer prediction was used.

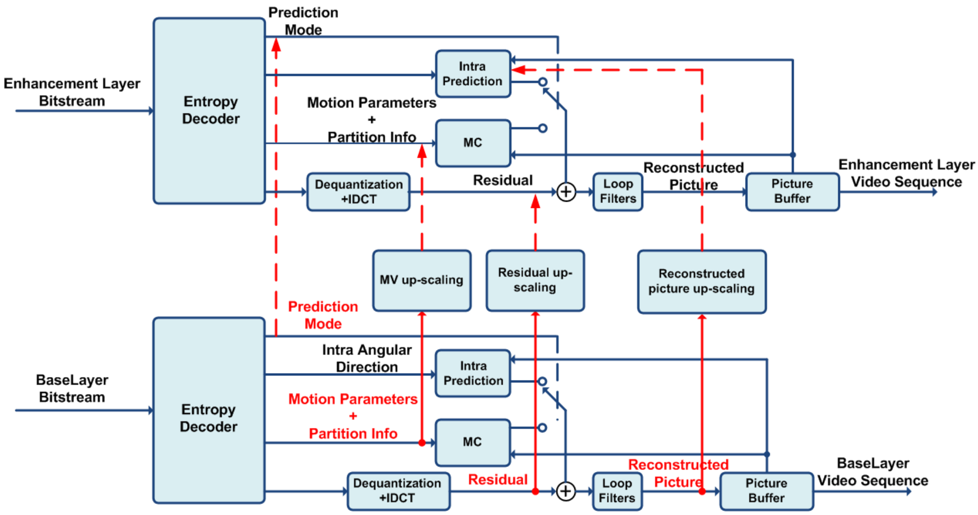


Figure 4-2 SVC test bed decoder structure

4.2 Inter Layer Intra Prediction

The idea of inter layer intra prediction is to use up-scaled base layer's reconstructed pixels for prediction. This prediction aims at exploiting the redundancy information between layer's pixel values.

The filter that is used for upscaling the reconstructed base layer's picture is the DCT-based interpolation filter (DCT-IF). This DCT-IF is used in HEVC for motion compensated prediction with half-pixel accuracy, as shown in Table 4-1. In H.264/SVC, up-sampling filters only have 4 taps for luma component and 2 taps for chroma component. Higher filter order could give the up-scaled base layer's pictures better approximation of enhancement layer's picture. Then, the enhancement layer's encoder tends to use the up-scaled base layer's reconstructed pixels for prediction which increases the coding efficiency. It has been tested that this DCT-IF gives additional 1%~2% gains compared to SVC up-sampling filters. But the computation complexity also goes up with the increasing of filter taps.

Table 4-1 DCT-IF coefficients

Luma	{ -1, 4, -11, 40, 40,-11,4,-1 }
Chroma	{ -4, 36, 36, -4 }

The filter operation is in two steps, the first step is vertical filtering. For example, as the A-H are original pixel values, the I pixel is evaluated as

$$\boldsymbol{I} = \left(-\boldsymbol{A} + 4\boldsymbol{B} - 11\boldsymbol{C} + 40\boldsymbol{D} + 40\boldsymbol{E} - 11\boldsymbol{F} + 4\boldsymbol{G} - \boldsymbol{H}\right)$$

The second step is horizontal filtering, i.e. to get pixels at L, J, K positions.

The syntax change in PU is shown in Table 4-2. use_base_layer_flag is in SPS header and is used for signaling whether inter layer intra prediction is used in high quality/resolution layer. Intra_bl_flag indicates whether using prediction from base layer reconstructed pixels or not. As shown in the syntax, the use of inter layer prediction is limited for PU size from 64x64 down to 8x8. When the enhancement layer's PU has size 4x4, it means the current block is likely to have much detailed information and it would be better to use prediction from neighbours at the same layer.

Three contexts models are used for coding of intra_bl_flag, these contexts are the same as for coding of base_mode_flag in H.264/SVC. The choice of context model (represented by context index *I*) depends on the neighbouring PU to the left (A) and PU on the top (B), as shown in Figure 4-3. The aim of using three contexts is to exploit statistical dependencies between neighbouring prediction information, e.g. if the neighbouring PUs are predicted from base layer, the current PU is likely to be predicted from the base layer. It has been tested that using three context models for coding this intra_bl_flag are better than directly coded using arithmetic coding.

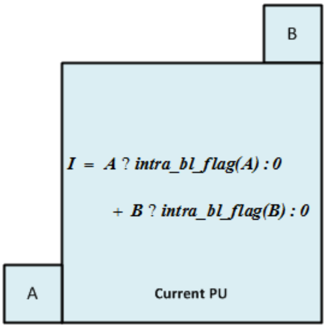


Figure 4-3 Context model of intra_bl_flag

Table 4-2 Syntax changes in Prediction Unit

prediction_unit(x0, y0, log2CUSize) {	**Descriptor**
if(skip_flag[x0][y0]) {	
…	
} else if(PredMode = = MODE_INTRA) {	
if(PartMode == PART_2Nx2N && pcm_enabled_flag && log2CUSize >= Log2MinIPCMCUSize && log2CUSize <= Log2MaxIPCMCUSize)	
pcm_flag	ae(v)
if(pcm_flag) {	
…	
} else {	
If(PartMode == PART_2Nx2N && **use_base_layer_flag**)	
intra_bl_flag	ae(v)
If(!intra_bl_flag)	
{	
...	
}	
}	
} else { /* MODE_INTER */	
…}	

In the HEVC encoder for enhancement layer, original encoding process is modified as in Figure 4-4. The inter layer intra prediction is integrated with normal MODE_INTRA decision. Generally, when the encoder chooses intra mode to encode a CU, there might be some new objects appeared in current picture or the video sequence is a fast moving sequence. It's reasonable to evaluate using prediction from the same block position in the up-scaled reconstructed base layer's picture before evaluate normal intra prediction. After inter layer intra prediction, the residuals are transformed and encoded. The transform process is similar to inter-coded CU's transform process since this inter layer intra prediction can be viewed as a special case of inter prediction. This inter layer intra prediction requires multi-loop decoding since the use of reconstructed base layer's pixels for prediction could also be used when the base layer's collocated CU is inter-coded.

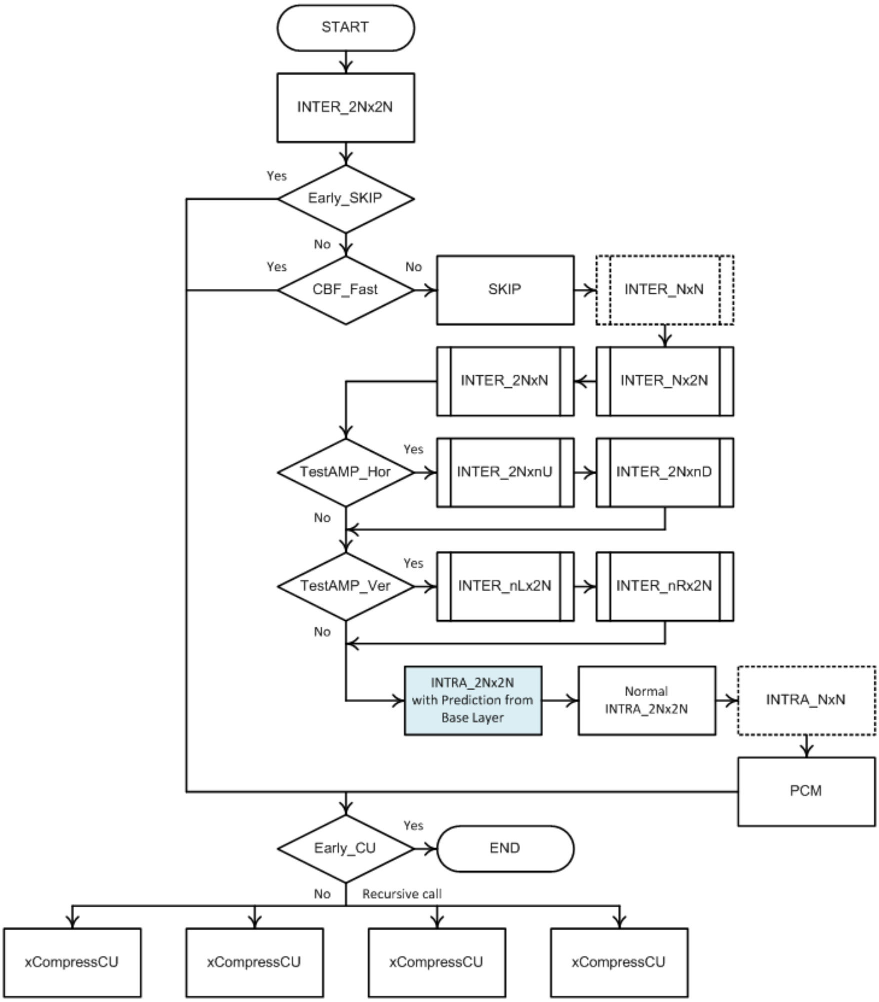


Figure 4-4 The scheme of modified encoder decision process

Results and Conclusions

The inter layer intra prediction capable HEVC encoder is tested with all intra high efficiency mode. The QP set for enhancement layer is [22, 27, 32, 37]. The result is shown in Table 4-3. When dQP is 0, the gain vs. simulcast is about 13.2% while the cost vs. single layer is about 17.5%. When dQP is 2, the gain vs. simulcast is about 16.5% while the cost vs. single layer is about 20.2%.

The results show that for high resolution like Class A and Class B sequences, the coding gain is higher than coding cost. The results are in line with expectation. The enhancement layer with high resolution has a base layer with better quality. The upscaled base layer could provide better prediction for coding enhancement layer. Moreover, some high frequency components in the video sequence would be missing after downsampled to create base layer but the upscaling filter could not reconstruct these high frequency components. The low resolution sequence suffers more from that.

Table 4-3 Inter layer intra prediction performance in intra HE

Cost: Single Layer vs. Scalable

	Intra HE (dQP = 0)		
	Y BD-Rate	U BD-Rate	V BD-Rate
Class A	14.3	18.9	15.5
Class B	13.2	21.7	19.1
Class C	19.3	24.3	26.2
Class D	20.4	26.3	27.5
Class E	20.9	23.6	24.8
All	17.5	23.3	23.1

Gain: Simulcast vs. Scalable

	Intra HE (dQP = 0)		
	Y BD-Rate	U BD-Rate	V BD-Rate
Class A	-20.4	-17.1	-19.5
Class B	-16.2	-10.0	-12
Class C	-9.2	-5.4	-4.0
Class D	-7.9	-3.3	-2.3
Class E	-15.9	-13.9	-13.1
All	-13.2	-8.9	-9.1

Cost: Single Layer vs. Scalable

	Intra HE (dQP = 2)		
	Y BD-Rate	U BD-Rate	V BD-Rate
Class A	16.4	20.5	16.7
Class B	15.3	24.2	21.2
Class C	22.3	28.6	30.1
Class D	23.7	31.2	32.6
Class E	23.1	25.2	26.7
All	20.2	26.5	26.4

Gain: Simulcast vs. Scalable

	Intra HE (dQP = 2)		
	Y BD-Rate	U BD-Rate	V BD-Rate
Class A	-24.5	-21.6	-24.1
Class B	-19.9	-13.8	-15.7
Class C	-12.1	-7.4	-5.8
Class D	-10.1	-4.5	-3.5
Class E	-20.0	-18.5	-17.6
All	-16.5	-12.0	-12.1

4.3 Inter Layer Motion Prediction

The idea of inter layer motion prediction is to reuse the base layer's motion parameters. This prediction aims at exploiting the redundancy information between layer's motion information. The enhancement layer's encoder was modified to incorporate base layer's motion parameters into motion merge prediction and base layer's motion vector into motion vector prediction.

The collocated base layer position (x', y') is derived from the down-scaled PU centre coordinate which is ((x + Lx) >> 1, (y + Ly) >> 1). If collocated PU is inter or skip coded, its reference picture list, reference picture index and up-scaled motion vector could be used for prediction of enhancement layer.

The modified HEVC encoder adds the collocated base layer's position into merge candidate list and replaces candidate B2 with the collocated base layer candidate BB, as shown in Figure 4-5. Different inserting positions had been tested and these tested positions were all before temporal candidate, e.g. position that after temporal candidate was not tested. Results show that replacing spatial above-left candidate with basic PU performs best.

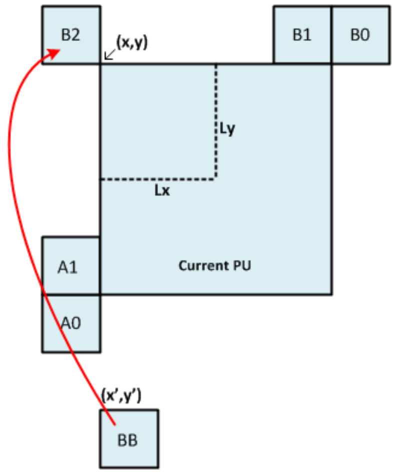


Figure 4-5 Derivation of motion merge candidates

The modified encoder also adds one scaled collocated base layer motion vector to the end of the motion vector prediction candidate list. It has been tested that adding the basic PU's motion vector to the end of motion vector prediction candidate list gives better gain than other position in the list. The derivation of collocated base layer position is the same as in merge candidate mentioned above. The scaling of base layer's motion vector is shown in Figure 4-6. D_EL is the distance between the current picture and EL's reference picture, D_BL is the distance between the current picture and BL's reference picture. The scaling factor is 2*(D_EL/D_BL), where the ratio D_EL/D_BL is to scale the BL's motion vector to point to the same reference picture and 2 is the ratio between EL and BL.

It's reasonable to add a PU's base layer's motion after its spatial candidates in both merge candidate list and motion vector prediction candidate list since the PU's spatial candidates are likely to have higher correlation with current PU's motion.

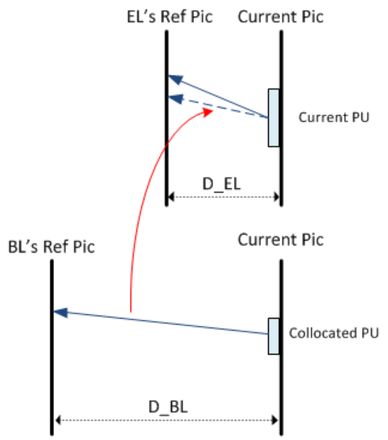


Figure 4-6 Scaling of base layer's motion vector

Results and Conclusions

The modified encoder without inter layer intra prediction is tested in low delay high efficiency mode. The QP set for enhancement layer is [22, 27, 32, 37], dQP between two layers is 2.

Table 4-4 Inter layer motion prediction in low delay coding

Gain: Simulcast vs. Scalable

	Low delay HE (dQP = 2)		
	Y BD-Rate	U BD-Rate	V BD-Rate
Class A	-	-	-
Class B	-1.0	-1.1	-1.3
Class C	-0.6	-0.6	-0.9
Class D	-0.4	-0.7	-0.6
Class E	-0.9	-1.1	-0.8
All	-0.7	-0.9	-0.9

The encoder's performance results with both inter layer intra prediction and inter layer motion prediction enabled are shown in Table 4-5. Most of the gain comes from multi-loop inter layer intra prediction. The inter layer motion prediction gives average of 0.7% and also shows there are some benefits using base layer's motion. But this 0.7% gain also indicates that the motion information from the base layer could not provide better prediction in most cases compared to that from neighbouring positions of the same layer. There also exists one big problem which limits this inter layer motion prediction's efficiency. As illustrated in figure below, the red line means the collocated base layer CU is subdivided into different

PUs. The motion parameter from position 4 is added to prediction candidate list according to the derivation process as above. But the motion vector from position 4 might be inappropriate for the whole CU/PU block of enhancement layer and it would not be chosen after RDO. In this way, this inter layer motion prediction's efficiency decreases.

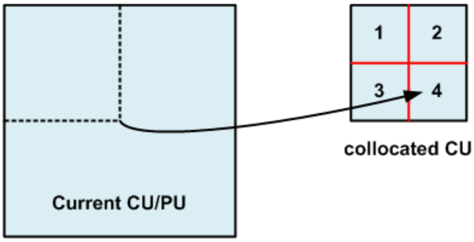


Table 4-5 Combined ILIP and ILMP performance

Cost: Single Layer vs. Scalable

	Low delay HE (dQP = 2)		
	Y BD-Rate	U BD-Rate	V BD-Rate
Class A	-	-	-
Class B	28.1	34.5	35.3
Class C	31.3	35.5	37.6
Class D	31.1	36.0	34.8
Class E	43.3	48.9	50.3
All	32.5	37.8	38.6

Gain: Simulcast vs. Scalable

	Low delay HE (dQP = 2)		
	Y BD-Rate	U BD-Rate	V BD-Rate
Class A	-	-	-
Class B	-10.3	-5.6	-5.1
Class C	-6.9	-3.7	-2.3
Class D	-4.1	-0.3	-1.2
Class E	-3.4	0.8	1.6
All	-6.6	-2.6	-2.1

Cost: Single Layer vs. Scalable

	Random Access HE (dQP = 2)		
	Y BD-Rate	U BD-Rate	V BD-Rate
Class A	29.6	50.3	44.8
Class B	20.7	36.9	35.4
Class C	27.4	36.1	38.6
Class D	28.1	37.4	37.4
Class E	-	-	-
All	25.6	38.6	38.0

Gain: Simulcast vs. Scalable

	Random Access HE (dQP = 2)		
	Y BD-Rate	U BD-Rate	V BD-Rate
Class A	-15.5	-2.1	-5.6
Class B	-15.2	-3.9	-4.8
Class C	-9.3	-3.0	-1.2
Class D	-6.0	0.9	0.9
Class E	-	-	-
All	-11.2	-2.1	-2.4

4.4 "H.264/SVC-like" Base Mode

The idea of base mode in current design is similar to that in H.264/SVC. When a CU is in base mode and its collocated base layer CU is inter-coded, its partition size, motion parameters are derived from this base CU (inter layer motion prediction). When the collocated base layer CU is intra-coded, the reconstructed pixels from this base CU are used for prediction (inter layer intra prediction).

The encoder with base mode requires "single-loop-like" decoding to decrease complexity. It only uses the pixels from BL's intra-coded CU but reconstruction of intra-coded CU may require reconstruction of neighbouring inter-coded CU's boundary pixels.

The modified encoding process is shown in Figure 4-10 and the syntax is shown in Table 4-6. base_mode_flag indicates whether the current CU is in Base Mode or not. After encoder checked R-D cost in skip mode, the encoder invokes Base Mode decision process (Figure 4-11). The base mode decision process compares enhancement layer's CU depth with collocated base layer's CU depth and decides whether the base mode is available for the enhancement layer's CU or not. The depth here means depth from largest coding unit, e.g. 64x64 corresponds to depth of 0. There are three scenarios and are explained separately below.

In the first scenario, the collocated CU's depth is not larger than enhancement layer's CU's depth. The enhancement layer's CU will fully fall into a block size not larger than ¼ of the base layer's CU size. Then it is reasonable to set the EL's CU's partition size to 2Nx2N. However, there are special cases that the BL's CU depth equals to EL's CU depth and the BL's CU partition is asymmetric partition. There are possibilities that the EL's CU falls into two different partitions, then the enhancement layer's CU should choose one partition between 2NxN, Nx2N. But for simplicity, the encoder sets the EL's CU partition to 2Nx2N also in this special case.

Figure 4-7 shows an EL's CU with depth 1 and its collocated BL's CU with depth 0, the EL's CU only corresponds to 1/16 of the BL's CU size after scaling. Then EL's CU could take partition size of 2Nx2N (32x32 in this case) since no sub-division will across its collocated region in BL's CU. Figure 4-8 shows an example of special case, an EL's CU and its collocated BL's CU depth both equal to 1 which corresponds to size of 32x32. The EL's CU corresponds to a 16x16 block in Base layer's CU. When the BL's CU is divided into 24x32 and 8x32 (partition type is nLx2N, red line) or 32x8 and 32x24 (partition type is 2NxnU, green line), EL's partition 2Nx2N would not be appropriate.

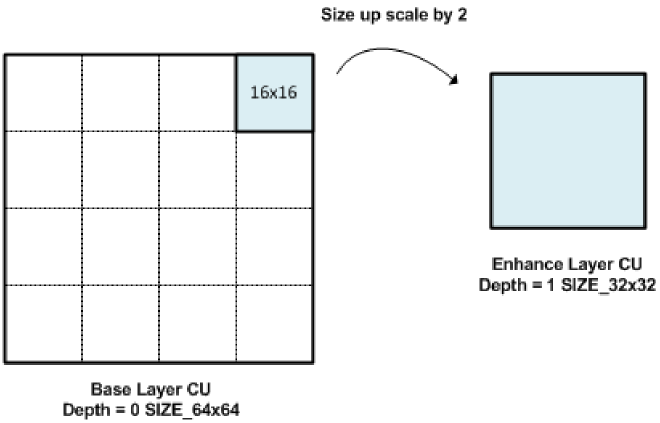


Figure 4-7 Example of relation of an EL's CU of depth 1 and its BL's CU of depth 0

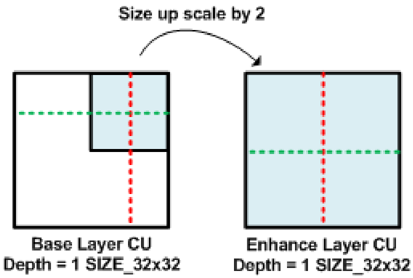


Figure 4-8 Example of relation of an EL's CU of depth 1 and its BL's CU of depth 1

The second scenario is the collocated BL's CU's depth equals to EL's CU depth + 1. In this case, the EL's CU will be fixed into the entire BL's CU so that the BL's CU's partition size could be directly used for EL's CU (Figure 4-9).

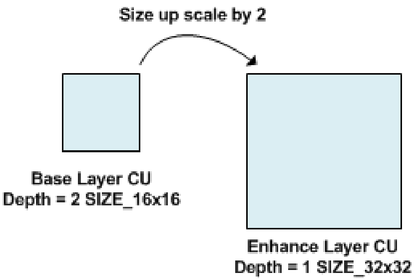


Figure 4-9 Example of relation of an EL's CU of depth 1 and its BL's CU of depth 2

The third scenario is the collocated CU's depth larger than EL's CU depth + 1. In this case, the EL's CU falls into different CU in BL and base mode would be unavailable for EL's CU.

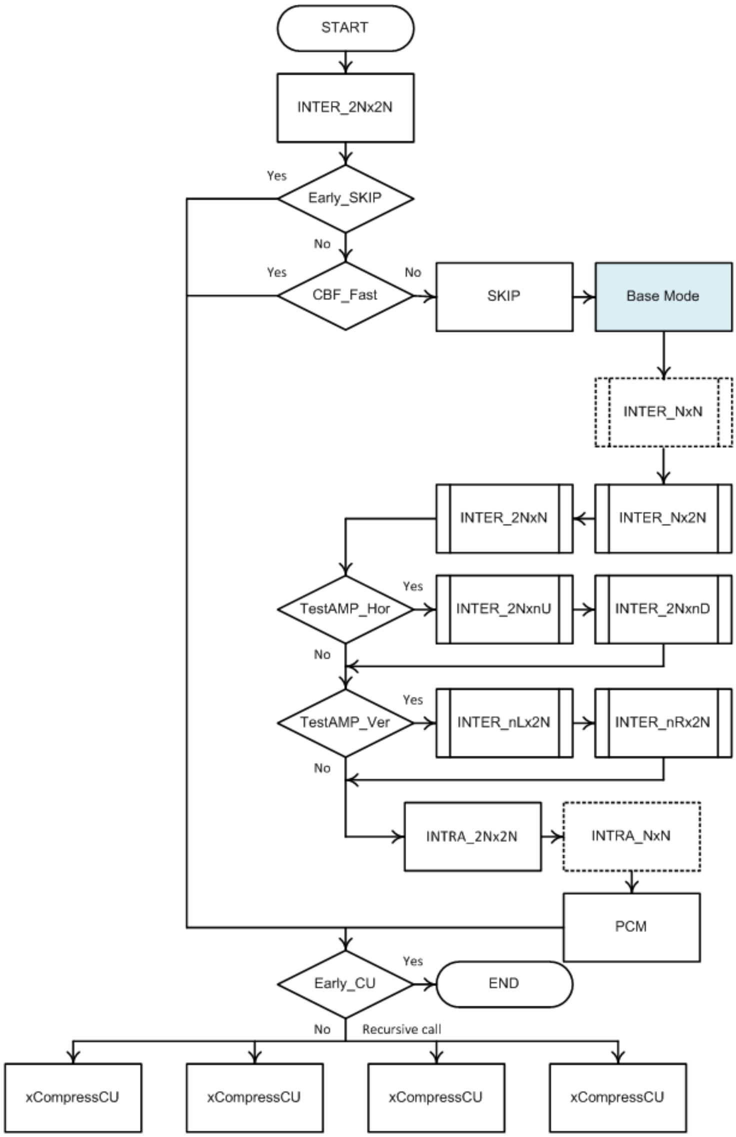


Figure 4-10 The scheme of encoder process when Base Mode is added

Table 4-6 CU syntax with base mode flag

coding_unit(x0, y0, log2CUSize) {	**Descriptor**
if(slice_type != I)	
skip_flag[x0 **][** y0 **]**	ae(v)
if(skip_flag[x0][y0])	
prediction_unit(x0, y0 , log2CUSize)	
else {	
base_mode_flag	
if(base_mode_flag)	
{	
prediction_unit(x0, y0 , log2CUSize);	
…	
}	
else if(slice_type != I || log2CUSize = = Log2MinCUSize) {	
…	
}	
if(!pcm_flag) {	
transform_tree(x0, y0, log2CUSize, log2CUSize, log2CUSize, 0, 0)	
transform_coeff(x0, y0, x0, y0, log2CUSize, log2CUSize, 0, 0)	
}	
}	
}	

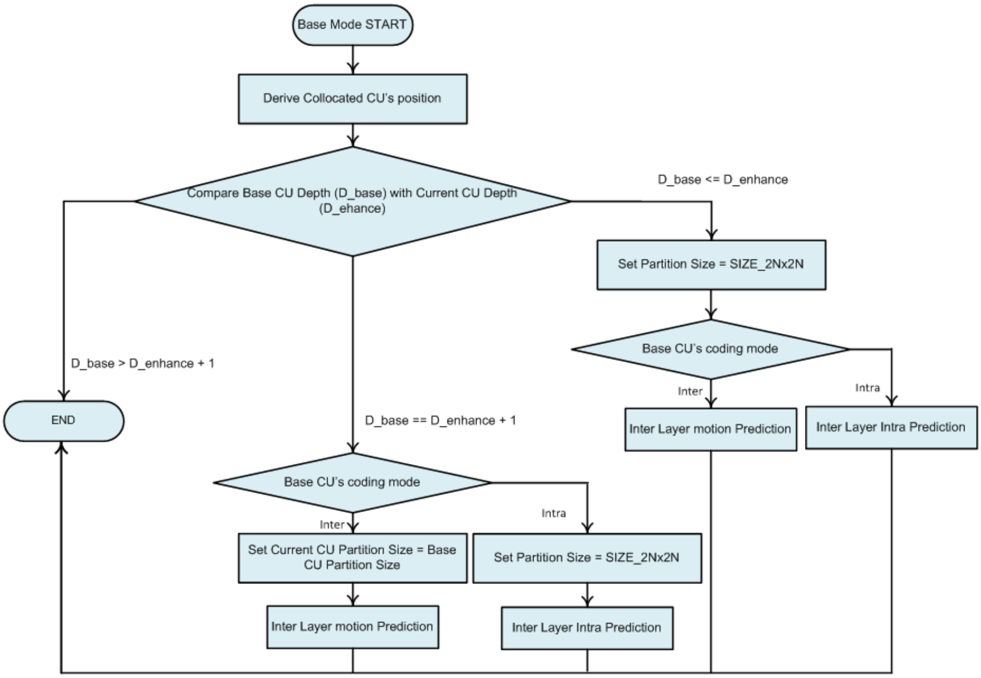


Figure 4-11 The scheme of Base Mode decision process

Results and Conclusions

The encoder with base mode enabled was tested in all-intra, low-delay and random-access mode. The QP set for enhancement layer is [22, 27, 32, 37], dQP between two layers is 2.

The results are shown in Table 4-7. In this base mode design, the base mode is quite limited since it largely depends on BL's CU depth. There might be a lot of CUs categorized into the third scenario and base mode would be unavailable for these CUs. For those base mode available CUs, even partition size from base layer might be suitable, there are possibilities that deriving its motion vectors from its spatial neighbours would give better prediction than directly using the upscaled motion vector from BL. The inappropriate upscaled base motion vector results in more cost for encoding residual information. Due to this reason, the base mode would be seldom chosen after RDO. Moreover, the "single-loop-like" decoding also degrades the performance.

Table 4-7 **Base mode performance results (dQP = 2)**

Cost: Single Layer vs. Scalable

	Intra HE (dQP = 2)		
	Y BD-Rate	U BD-Rate	V BD-Rate
Class A	16.2	21.5	17.7
Class B	15.3	24.7	22.4
Class C	22.3	29.1	31.4
Class D	23.5	31.4	32.7
Class E	22.9	26.1	24.8
All	20.1	27.0	27.2

Gain: Simulcast vs. Scalable

	Intra HE (dQP = 2)		
	Y BD-Rate	U BD-Rate	V BD-Rate
Class A	-24.5	-21.0	-23.4
Class B	-20.0	-13.5	-15.1
Class C	-12.1	-7.1	-5.4
Class D	-10.3	-4.3	-3.5
Class E	-20.1	-18.0	-17.2
All	-16.6	-11.6	-11.6

Cost: Single Layer vs. Scalable

	Low delay HE (dQP = 2)		
	Y BD-Rate	U BD-Rate	V BD-Rate
Class A	-	-	-
Class B	39.5	43.3	43.2
Class C	38.2	39.8	40.0
Class D	35.3	36.1	36.3
Class E	48.0	51.4	54.8
All	39.7	42.1	42.8

Gain: Simulcast vs. Scalable

	Low delay HE (dQP = 2)		
	Y BD-Rate	U BD-Rate	V BD-Rate
Class A	-	-	-
Class B	-2.6	0.3	0.2
Class C	-2.1	-0.7	-0.6
Class D	-1.1	-0.2	-0.1
Class E	-0.4	2.5	4.7
All	-1.7	0.3	0.8

Cost: Single Layer vs. Scalable

	Random Access HE (dQP = 2)		
	Y BD-Rate	U BD-Rate	V BD-Rate
Class A	37.7	52.8	48.9
Class B	28.3	41.8	38.9
Class C	32.1	38.8	39.9
Class D	31.2	37.3	37.6
Class E	-	-	-
All	31.3	41.3	40.1

Gain: Simulcast vs. Scalable

	Random Access HE (dQP = 2)		
	Y BD-Rate	U BD-Rate	V BD-Rate
Class A	-10.2	-0.3	-2.8
Class B	-10.0	-0.4	-2.3
Class C	-6.1	-1.1	-0.3
Class D	-3.9	0.8	1.1
Class E	-	-	-
All	-7.4	-0.3	-0.9

In order to compare with H.264/SVC, the CU size was limited to 16x16 and 8x8. TU size could choose from 16x16 down to 4x4 and AMP was turned off. Class E was tested for low delay configuration and Class B was tested for random access configuration. Results (Table 4-8) show that the gain is 1.6% for low delay and 11.7% for random access. If Motion Merge is disabled, the gain becomes 7.5% for low delay and 15.3% for random access, as shown in Table 4-9. This indicates that the reuse of enhancement layer's neighbouring motion information is more efficient than that of motion parameters from base layer. The motion merge enables enhancement layer's PUs to choose among five spatial and temporal candidates and that exploits the neighbouring redundancy quite a lot. While in H.264/SVC, the encoder could only choose motion vectors from one of its two spatial neighbouring inter coded CU or the median of neighbouring inter coded CU's motion vectors or from its temporal co-located inter coded CU. This doesn't make the full use of neighbouring redundancy information and there are higher possibilities that the base layer's motion parameters could give better prediction.

Finally, the limitation on CU size has effects on performance but not comparable to that results from motion merge.

Table 4-8 Cost and Gain for H.264/SVC like HEVC (Motion Merge On)

Cost: Single Layer vs. Scalable

	Low delay		
Class E	Y BD-Rate	U BD-Rate	V BD-Rate
Vidyo1_720p60	44.2	46.3	51.3
Vidyo3_720p60	43.4	42.5	53.9
Vidyo4_720p60	40.8	45.5	39.3
All	42.8	44.8	48.2

Gain: Simulcast vs. Scalable

	Low delay		
Class E	Y BD-Rate	U BD-Rate	V BD-Rate
Vidyo1_720p60	-1.5	0.1	3.7
Vidyo3_720p60	-1.1	-1.1	6.4
Vidyo4_720p60	-2.1	1.4	-2.6
All	-1.6	0.2	2.5

Cost: Single Layer vs. Scalable

	Random Access		
Class B	Y BD-Rate	U BD-Rate	V BD-Rate
Kimono1_1920x1072_24	23.5	38.8	34.6
ParkScene_1920x1072_24	26.1	44.1	30.9
Cactus_1920x1072_50	29.3	38.3	41.5
BQTerrace_1920x1072_60	20.1	30.0	28.7
BasketballDrive_1920x1072_50	24.4	33.2	33.6
All	24.7	36.9	33.9

Gain: Simulcast vs. Scalable

	Random Access		
Class B	Y BD-Rate	U BD-Rate	V BD-Rate
Kimono1_1920x1072_24	-18.4	-8.5	-11.3
ParkScene_1920x1072_24	-10.9	1.9	-7.1
Cactus_1920x1072_50	-10.5	-3.7	-1.7
BQTerrace_1920x1072_60	-4.6	3.1	2.0
BasketballDrive_1920x1072_50	-13.8	-7.6	-7.2
All	-11.7	-3.0	-5.0

Table 4-9 Cost and Gain for H.264/SVC like HEVC (Motion Merge Off)

Cost: Single Layer vs. Scalable

	Low delay		
	Y BD-Rate	U BD-Rate	V BD-Rate
Vidyo1_720p60	32.0	35.3	42.2
Vidyo3_720p60	36.9	36.2	46.8
Vidyo4_720p60	30.3	37.4	32.9
All	33.1	36.3	40.6

Gain: Simulcast vs. Scalable

	Low delay		
	Y BD-Rate	U BD-Rate	V BD-Rate
Vidyo1_720p60	-8.9	-6.5	-1.6
Vidyo3_720p60	-4.9	-4.9	2.2
Vidyo4_720p60	-8.7	-3.5	-6.5
All	-7.5	-5.0	-2.0

Cost: Single Layer vs. Scalable

	Random Access		
Class B	Y BD-Rate	U BD-Rate	V BD-Rate
Kimono1_1920x1072_24	12.2	34.8	28.8
ParkScene_1920x1072_24	24.2	43.2	29.9
Cactus_1920x1072_50	22.5	33.9	36.4
BQTerrace_1920x1072_60	18.8	29.8	28.9
BasketballDrive_1920x1072_50	18.6	29.7	29.3
All	19.3	34.3	30.6

Gain: Simulcast vs. Scalable

	Random Access		
Class B	Y BD-Rate	U BD-Rate	V BD-Rate
Kimono1_1920x1072_24	-25.6	-10.8	-14.8
ParkScene_1920x1072_24	-12.3	1.3	-7.8
Cactus_1920x1072_50	-14.9	-6.7	-5.1
BQTerrace_1920x1072_60	-6.0	2.6	1.9
BasketballDrive_1920x1072_50	-17.6	-9.7	-10.0
All	-15.3	-4.6	-7.1

4.5 Comparison with H.264/SVC and other arts on HEVC scalable extension proposal

In this thesis, the idea of inter layer intra prediction, base mode are similar to H.264/SVC. But in inter layer motion prediction is a bit different from H.264/SVC. The motion parameters from base layer are added to the enhancement layer's motion candidate list. The encoder chooses the base layer motion to encode only when the base layer motion gives better prediction than other candidate from the same enhancement layer.

There is a proposal from vidyo [9] on HEVC scalable extension. They introduced a new CU prediction mode called difference mode. When difference mode is enabled, the difference between the enhancement layer's pixels and upsampled base layer's pixels are encoded using normal HEVC. Their design shows 17.3% gain and 19.9% cost for intra HE, 15.5% gain and 21.1% cost for random access HE, 11.2% gain and 26.9% cost for low delay. In intra case, they used normal HEVC angular coding method to encode the residual signal further but it might be efficient enough to transform and encode the residual signal directly just as in my design. For inter cases, their encoder requires three motion compensation loops for complete reconstruction of base layer, encoding of inter-coded CU in normal mode and encoding of inter-coded CU in difference CU mode. That increases the performance a lot but introduces huge complexity both in encoder and decoder side.

CHAPTER 5

5 Conclusions

5.1 Achievements

In this thesis, inter layer intra prediction, inter layer motion prediction and base mode have been implemented and evaluated on top of HEVC. Inter layer residual prediction is not implemented due to the limitation of time.

In inter layer intra prediction design, the encoder could achieve 16.5% gain and 20.2% cost for all-intra HE case. In inter layer motion prediction, the motion parameters from base layer are added to the enhancement layer's candidate list. The results show about 0.8% gain. When combined the inter layer motion prediction and inter layer motion prediction together, the encoder shows 16.5% gain and 20.2% gain for all-intra HE case, 6.6% gain and 32.5% cost for low delay HE, 11.2% gain and 25.6% cost for random access HE. It's expected that most of the gain in inter case are coming from multi-loop inter layer intra prediction.

In the base mode design, the "single-loop-like" encoder gives 1.7% gain and 39.7% cost for low delay HE case, 7.4% gain and 31.3% cost for random access case. The "H.264/SVC–like" HEVC test was done to investigate the reason why coding gain was small for inter cases. The results indicate that the reuse of base layers motion parameters is not as efficient as that in H.264/SVC since the enhancement layer could get better predictions from information coming from same layer.

5.2 Future work

The SVC encoder in HEVC need further improvements on coding gain for inter cases while keeping an acceptable cost and complexity.

Inter layer residual prediction tools on top of HEVC worth evaluating. In SVC, ILRP outperforms ILMP, so the performance is expected to be better if ILRP is added with ILMP.

Besides the three inter layer prediction tools, new inter layer prediction tools need to be developed in order to achieve better performance. There might be gains if the enhancement layer could copy the whole base layer's coding unit tree structure as well as the transform tree structure. The layers' loop filter coefficients might also contain some redundancy information which could be exploited further.

List of Figures

List of Tables

Abbreviations

ALF	Adaptive Loop Filter
AMP	Asymmetric Motion Partition
AVC	Advanced Video Coding
BL	Base Layer
BD-Rate	Bjøntegaard-Delta Rate
CABAC	Context Adaptive Binary Arithmetic Coding
CU	Coding Unit
DCT	Discrete Cosine Transform
EL	Enhancement Layer
HEVC	High Efficiency Video Coding
HM	HEVC reference software
HVS	Human Visual System
ILIP	Inter Layer Intra Prediction
ILMP	Inter Layer Motion Prediction
ILRP	Inter Layer Residual Prediction
LCU	Largest Coding Unit
MC	Motion Compensation
ME	Motion Estimation
MPEG	Motion Picture Experts Group
MSE	Mean Square Error
PSNR	Peak Signal to Noise Ratio
PU	Prediction Unit
QP	Quantization Parameters
RDO	Rate Distortion Optimization
SAD	Sum of Absolute Difference
SAO	Sample Adaptive Offset
SATD	Hadamard Transformed SAD
SSE	Sum of Square Error
SVC	Scalable Video Coding
TU	Transform Unit
VCEG	Video Coding Expert Group

Bibliography

[1]. ITU-T Q6/16 Visual Coding and ISO/IEC JTC1/SC29/WG11 Coding of Moving Pictures and Audio, "Joint Call for Proposals on Video Compression Technology", *VCEG-AM91*, VCEG Meeting Kyoto, January 2010.

[2]. Benjamin Bross, Woo-Jin Han, Jens-Rainer Ohm, Gary J. Sullivan, and Thomas Wiegand, "High Efficiency Video Coding (HEVC) text specification Working Draft 5", *JCTVC-G1102*, JCT-VC Meeting Geneva, November 2011.

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

[4]. G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves", *VCEG-M33*, VCEG Meeting Austin, April 2001.

[5]. Frank Bossen, "Common test conditions and software reference configurations", *JCTVC-G1200*, JCT-VC Meeting Geneva, November 2011.

[6]. Ken McCann, Benjamin Bross, Il-Koo Kim, Shun-ichi Sekiguchi, and Woo-Jin Han, "High Efficiency Video Coding (HEVC) Test Model 5 Encoder Description", *JCTVC-G1103*, JCT-VC Meeting Geneva, November 2011.

[7]. Detlev Marpe, Heiko Schwarz, and Tomas Wiegand, "Context-Based Adaptive Binary Arithmetic Coding in the H.264/AVC Video Compression Standard", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol.13, No.7, July 2003.

[8]. Heiko Schwarz, Detlev Marpe, and Tomas Wiegand, "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard", *IEEE Transactions on Circuits and Systems for Video Technology,* Vol.17, No.9, September 2007.

[9]. Danny Hong, Wonkap Jang, Jill Boyce and Adeel Abbas, "Scalability Support in HEVC", *JCTVC-F290*, JCT-VC Meeting Torino, July, 2011