Complexity Reduction Algorithm for Quality Scalability in Scalable HEVC

[1]Yuan-Shing Chang, [1]Ke-Nung Huang and [*,1]Chou-Chen Wang

Abstract

SHVC, the scalable extension of high efficiency video coding (HEVC), can improve the compression performance by using advanced inter-layer prediction features at the cost of huge computational complexity. Recently, in order to reduce the encoding complexity of SHVC, a tempo-spatial searching order algorithm (TSSOA) and a fast CU depth range decision (FCUDRD) are proposed, respectively. However, every coding unit (CU) in these two methods still need perform motion estimation (ME) to find the best prediction mode. To further improve the performance of TSSOA and FCUDRD, we propose two encoding strategies including fast prediction unit (PU) prediction algorithm (FPUPA) and fast motion vector (MV) prediction algorithm (FMVPA) in this paper. Firstly, we use TSSOA to find the best candidate quadtree, and then the neighboring PU modes are considered as the best prediction mode of the current CU. Secondly, five causal neighboring MVs of the CUs are considered as the good candidate MV of the current CU due to temporal and spatial correlation of MV. Finally, we combine FPUPA and FMVPA into the SHVC system to further speed up the encoding process. Simulation results show that the proposed FPUPA and FMVPA can achieve an average of time improving ratio (TIR) about 69.39% and 71.70% for LD, when compared to original SHVC (SHM4.0). In addition, as compared with TSSOA and FCUDRD, the proposed method can further achieve an average of TIR about 13.52% and 14.35%, respectively. It is clear that the proposed algorithm can efficiently increase the speed of SHVC encoder with insignificant loss of image quality.

Keywords: video coding standard, HEVC, Scalable HEVC, motion estimation

*Corresponding Author: Chou-Chen Wang
(E-mail: E-mail:chchwang@isu.edu.tw)
[1] Department of Electronic Engineering, I-Shou University, Kaohsiung, Taiwan

1. Introduction

Nowadays, high definition (HD) video applications have become part of our everyday lives. In the other hand, ultra HD (4K×2K or 8K×4K) contents have already become relatively popular in commercial applications and begin to attract the market's attention. However, the current video coding standard H.264/AVC [1-2] is difficult to meet the emerging demands of HD and UHD resolutions. Therefore, the ITU-T and ISO/IEC through their Joint Collaborative Team on Video Coding (JCT-VC) has developed a newest high efficiency video coding (HEVC) for the video compression standard to satisfy the UHD requirement in 2010, and the first version of HEVC was approved as ITU-T H.265 and ISO/IEC 23008-2 by JCT-VC in Jan. 2013 [3-4]. HEVC can achieve an average bit rate decrease of 50% in comparison with H.264/AVC while still maintaining video quality. This is because the HEVC adopts some new coding structures including coding unit (CU), prediction unit (PU) and transform unit (TU). The HEVC adopts the quadtree-structured coding tree unit (CTU), and each CTU allows recursive splitting into four equal sub-CUs. The HEVC can achieve the highest coding efficiency, but it requires a very high computational complexity such that its real-time application is limited. In addition, most of HD and UHD video applications involve different devices which have different screen resolutions, CPU processing capabilities and network bandwidth requirements. Thus, to further upgrade the HEVC used in heterogeneous access networks, the JVT-CT develops a scalable HEVC (SHVC), and was finalized in July 2014 [5-7]. The function of SHVC includes spatial scalability, temporal scalability and SNR scalability. Based on the HEVC, the SHVC scheme supports multi-loop solutions by enabling different inter-layer prediction mechanisms. Although the SHVC can achieve the highest coding efficiency, it requires a higher computational complexity than HEVC. As a result, the high computational complexity of SHVC has become an obstruction for the real-time services.

In order to reduce the computational complexity of SHVC encoder, recently, a tempo-spatial searching order algorithm (TSSOA) [8] was proposed for quality scalability to find a good candidate quadtree of the current CTU in SHVC. On the other hand, Fast CU depth range decision (FCUDRD) algorithms based on the maximal and minimal values of depth levels to determine current CU depth were presented in [9-10]. However, every CU splitting process by TSSOA and FCUDRD methods still need perform ME module, leading to a waste of encoding time. To further improve the performance of TSSOA and FCUDRD, we propose two encoding strategies including fast PU prediction algorithm (FPUPA) and fast MV prediction algorithm (FMVPA). Firstly, we use TSSOA to find the best candidate quadtree, and then the PU neighboring modes are considered as the best prediction mode of the current CU. Secondly, five causal neighboring MVs of the CUs are considered as the good candidate MV of the current CU due to tempo-spatial correlation of MV. Finally, we combine FPUPA and FMVPA into the SHVC system to further speed up the encoding process.

The rest of the paper is organized as follows. Section 2 gives some overviews of HEVC and SHVC. In Section 3 we describe the proposed fast PU decision algorithm for SHVC encoder. The results of experiment are shown in Section 4. Finally, Section 5 shows the conclusion of this study.

2. Overviews of HEVC and SHVC

HEVC can greatly improve coding efficiency by adopting hierarchical structures of CU, PU and TU [3]. In general, the CU depths can be split by coding the quadtree structure of 4 levels, and the CU size can vary from the largest CU (LCU: 64×64) to the smallest CU (SCU: 8×8). The CTU is the largest CU. During the encoding process, each CTU block of HEVC can be split into four equally sized blocks according to inter/intra prediction in rate-distortion optimization (RDO) sense. At each depth level of CTU, HEVC performs motion estimation and compensation (ME/MC), transforms, and quantization with different size. The PU module is the basic unit used for carrying the information related to the prediction processes, and the TU can be split by residual quadtree (RQT) maximally at 3 level depths which vary from 32×32 to 4×4 pixels.

In general, intra-coded CUs have only two PU partition types including 2N×2N and N×N, but inter-coded CUs have eight PU types including symmetric blocks (2N×2N, 2N×N, N×2N, N×N) and asymmetric blocks (2N×nU, 2N×nD, nL×2N, nR×2N) [3-4]. When only using symmetric PU blocks, HEVC encoder tests 7 different partition sizes including SKIP, inter 2N×2N, inter 2N×N, inter N×2N, inter N×N, intra 2N×2N and intra N×N for an inter slice as shown in Figure 1. The rate distortion costs (RDcost) have to be calculated by performing the PUs and TUs to select the optimal partition mode under all partition modes for each CU size. The encoding and pruning procedure of a CTU is demonstrated in Figure 2. Since all the PUs and available TUs have to be exhaustively searched by RDO process for an LCU, HEVC dramatically increases computational complexity compared with H.264/AVC. The optimization of the block mode decision procedure will result in the high computational load and limit the use of HEVC encoders in real-time applications.

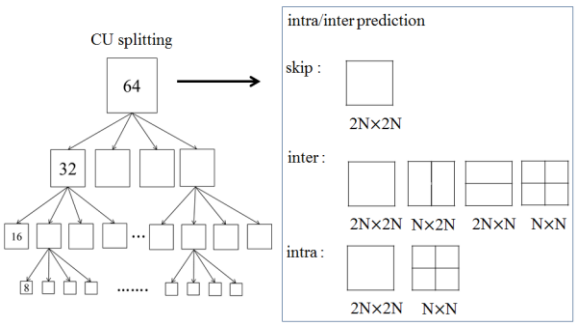


Figure 1: Recursive CU splitting for skip, inter and intra modes in PU module.

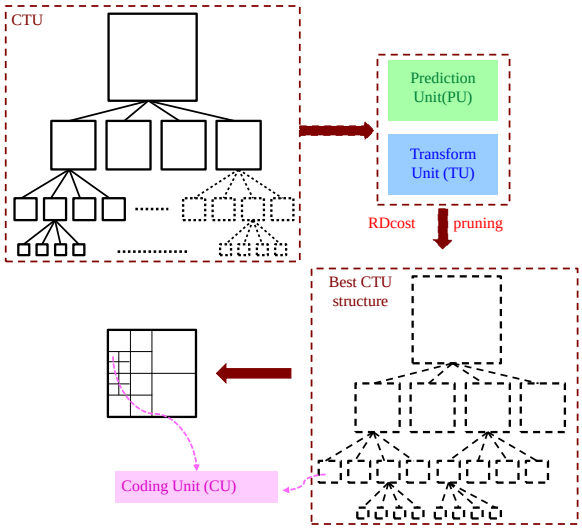


Figure 2: The encoding and pruning procedure of a CTU.

Since the coding procedure for HEVC is very complex, this leads to a much more complex encoding process of SHVC due to an extension of HEVC.

Based on HEVC, the SHVC scheme supports both single-loop and multi-loop solutions by enabling different inter-layer prediction mechanisms [11-13]. A typical architecture of two layers SHVC encoder including base layer (BL) and enhancement layer (EL) is shown in Fig. 3. However, the SHVC encoder allows one BL and more than one EL. For spatial

scalability, the input high-resolution video sequence should be down-sampled to get the low-resolution video sequence, but for SNR scalability BL and EL layer uses the same resolution video sequence. Therefore, there are larger redundancies between different layers for quality/SNR (signal-to-noise ratio) scalability.

The inter prediction and intra prediction modules of the EL encoder are modified to accommodate the BL pixel samples in the prediction process. The BL syntax elements including motion parameters and intra modes are utilized to predict the corresponding EL syntax elements to reduce the overhead for coding syntax elements. Furthermore, the transform/quantization and inverse transform/inverse quantization modules (denoted as DCT/Q and IDCT/IQ, respectively, in Figure 3) are applied to inter-layer prediction residues for better energy compaction. From [11-12], we can find that the computational complexity of HEVC encoder is several times more than H.264/AVC encoder. As the extension of HEVC, SHVC encoder is expected to be several times more than HEVC encoder depending on the number of EL. Therefore, the study on how to reduce the computational complexity of SHVC encoder is an important subject especially for expecting to achieve real-time HD/UHD video applications in heterogeneous access networks.

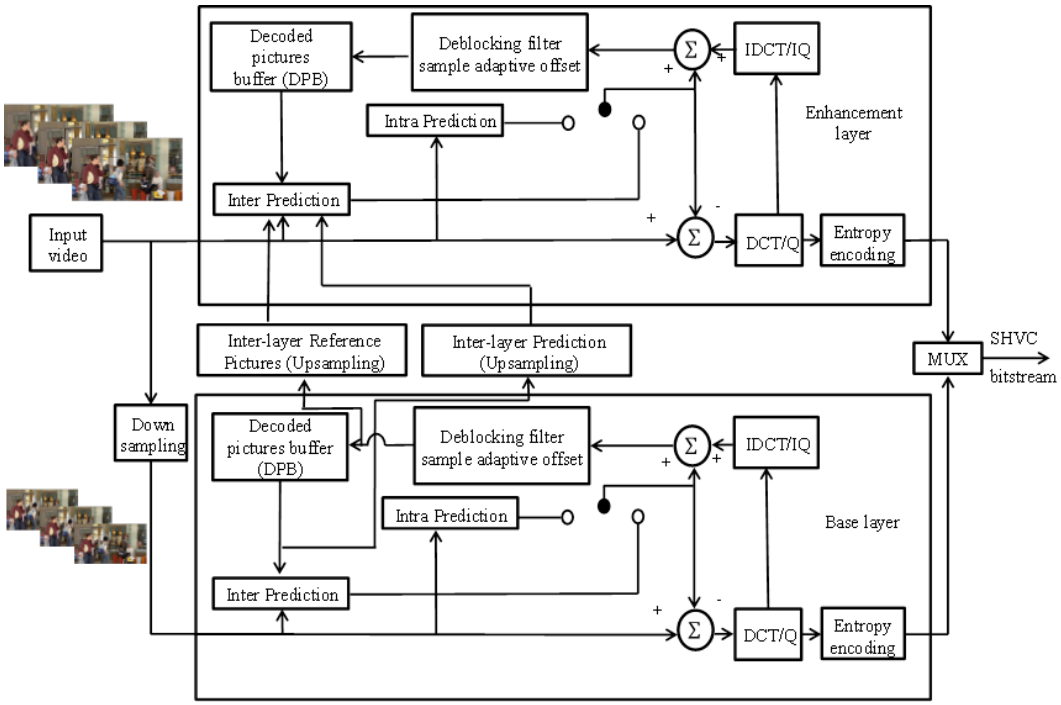


Figure 3: A typical architecture of two layers SHVC encoder.

3. Proposed Fast PU Decision Algorithm

To reduce the computational complexity of SHVC encoder, TSSOA [8] and FCUDRD [9] are two very effective methods recently. However, every CU selected by these two methods still need perform ME module to find the best PU mode. This leads to decrease the speedup of SHVC encoding. In order to further improve the performance of SHVC, we propose a complexity reduction algorithm by a combination of TSSOA and FCUDRD.

3.1 Previous Fast Algorithms

1) TSSOA

TSSOA mainly utilizes the characteristics of natural video sequences existing strongly temporal and spatial correlation to speed up the encoding process of SHVC. In this work, five causal neighboring split CTUs are first sequentially selected to find the best candidate according to the searching order decided by the sort of probability values in BL and EL. Figure 4 shows the corresponding five causal encoded neighboring CTUs(B_A~B_E) of the current CTU(B_X) in the tempo-spatial direction in BL, respectively. Figure 5 shows the search priority order in BL according to the correlation values determined by experiments. Block 1 represents the temporal neighbor, and blocks 2 to 5 denote spatial neighbors in horizontal, vertical, 45 and 135 diagonal directions.

To determine whether a candidate split structure of the CTU is good enough for the current CTU, TSSOA checks computing the RD cost by using the predicted split structure. After the candidate split structure (one of blocks 1 to 5) is found, TSSOA checks whether it is good enough for the current CTU by comparing its RD cost with a threshold. If it is less than the threshold, the candidate is good enough for the current CTU. Otherwise, it implies that the tempo-spatial correlation is low, and a full recursive process is needed to find the optimal split structure of the current CTU. For fast EL encoding, TSSOA uses the inter-layer searching order algorithm (ILSOA) between BL and EL as the same as TSSOA to predict the split quadtree of CTU for the current CTU in EL. ILSOA shows the encoding performance with negligible decrease when only utilizing three candidates in EL as shown in Figure 6. TSSOA can implement an early termination (ET) split quadtree search, so the encoder does not need to go through all the modes, thus significantly reducing the computational complexity.

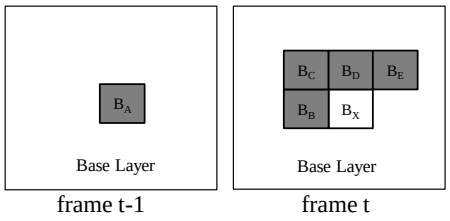


Figure 4: Corresponding five causal encoded neighboring CTUs in the tempo-spatial direction in BL.

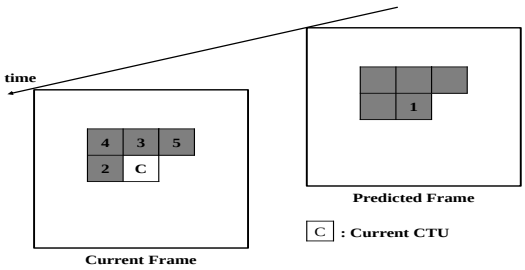


Figure 5: The search priority order in BL.

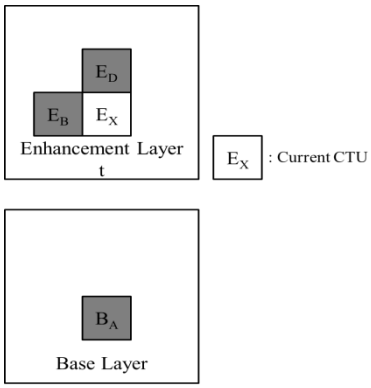


Figure 6: Three causal encoded neighboring CTUs as candidates in EL.

TSSOA for SHVC encoder can be summarized as follows:

Step 1. Set a threshold (Thr_{QP_tree}) value according to QP.

Step 2. Encode the BL of SHVC using TSSOA. If the RDcost computed by priority 1 is less than Thr_{QP_tree} ,go to step 6. Otherwise, go to step 3.

Step 3. If it is the last neighboring CTU, go to step 5. Otherwise, go to step 4.

Step 4. Compute RDcost of next neighboring CTU in the searching order (2~5); if the RDcost is less than Thr_{QP_tree},, go to step 6. Otherwise, go to step 3.

Step 5. Use the original RDO module to prune the best quadtree of current CTU.

Step 6. Record the best CTU quadtree and parameters of BL.

Step 7. It is similar to encode BL. Encode the EL of SHVC using ILSOA as performing steps of TSSOA in BL.

Step 8. Record the best CTU quadtree and parameters of EL.

2) *FCUDRD*

Based on the depth information correlation between tempo-spatial adjacent CTUs and the current CTU, FCUDRD is adaptively excludes from the depth search process in advance. The best depth of current CTU is determined by an intersection between temporal predicted depth ranges by 9 Gaussian weighting from encoded blocks and spatial predicted depth ranges by 4 best weighting from encoded blocks, separately. The optimal CTU depth level of a block by using FCUDRD is predicted using tempo-spatial neighboring blocks. The temporal predicted depth is defined as follows:

$$Depth_{temp\_pred} = \sum_{i=0}^{N-1} \omega_i \cdot d_i \quad (1)$$

where N is the number of previous encoded CTUs and is equal to 9, d_i is the value of depth level, and ω_i is the weight determined based on correlations between the current CTU and its temporal neighbouring CTUs as shown in Figure 7. On the other hand, the spatial predicted depth is defined as follows:

$$Depth_{spat\_pred} = \sum_{i=0}^{M-1} \alpha_i \cdot d_i \qquad (2)$$

where M is the number of previous encoded CTUs and is equal to 4, d_i is the value of depth level and α_i is the weight determined based on correlations between the current CTU and its spatial neighbouring CTUs as shown in Figure 7.

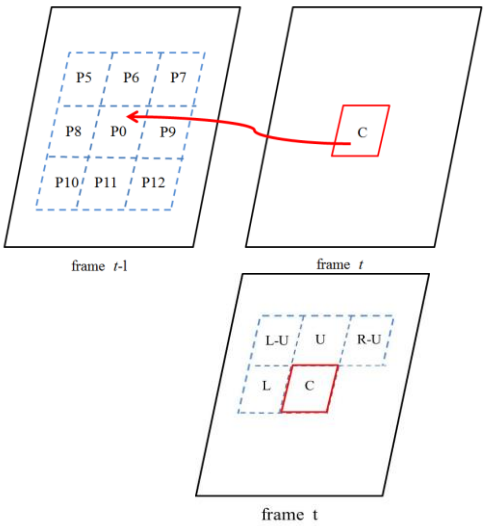


Figure 7: tempo-spatial adjacent CTUs of the current CTU. C: current CTU; L: left CTU; L-U: left upper CTU; U: upper CTU; R-U: right upper CTU; P0: co-located CTU; P5~P12: temporally co-located neighboring CTUs.

The best predicted depth range of current CTU is determined from the intersection of two predicted depth ranges as follows:

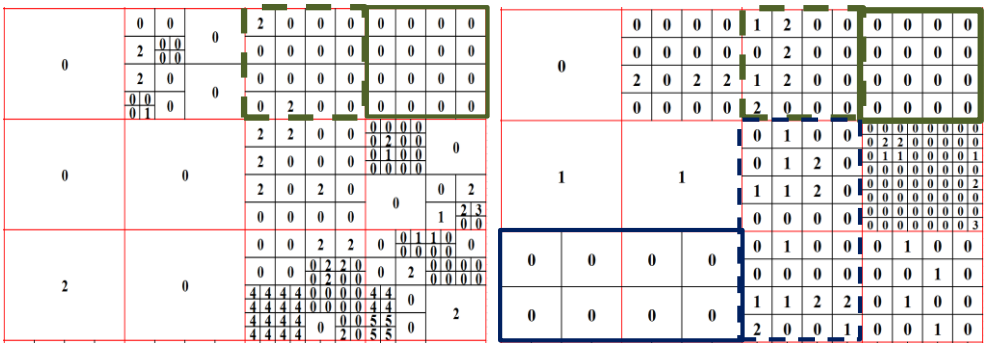
$$Depth_{best\_pred} = Depth_{temp\_pred} \cap Depth_{spat\_pred} \qquad (3)$$

According to the predicted value of the best CTU depth, each block is divided into five types as follows:

(1) If $Depth_{best\_pred} = 0$, its best CTU depth is chosen to "0". The dynamic depth range (DDR) of current CTU is classified as Type 0.
(2) If $0 < Depth_{best\_pred} \leq 0.5$, its best CTU depth is chosen to "0". The DDR of current CTU is classified as Type 1.
(3) If $0.5 < Depth_{best\_pred} \leq 1.5$, its best CTU depth is chosen to "1", The DDR of current CTU is classified as Type 2.
(4) If $1.5 < Depth_{best\_pred} \leq 2.5$, its best CTU depth is chosen to "2". The DDR of current CTU is classified as Type 3.
(5) If $Depth_{best\_pred} > 2.5$, its best CTU depth is chosen to "3". The DDR of current CTU is classified as Type 4.

3.2. Proposed FPUPA

Although TSSOA for quality scalability can find a good candidate quadtree of the current CTU, the PU partition modes are not considered to exploit the correlation existing images. Therefore, there still are some extra computational loads which have not been fully employed. To further reduce computational calculation of TSSOA, we firstly observe and carry the statistic distribution of optimal PU modes using a quantization parameter pair (QP_{BL}, QP_{EL})=(32, 28) under the test model SHM 4.0 [14]. For convenience of PU partition mode description, we set PU mode numbers of 0, 1, 2, 3, 4 and 5 to represent mode inter 2N×2N, inter 2N×N, inter N×2N, inter N×N, intra 2N×2N and intra N×N, respectively.



Predicted Frame(BL) Current Frame(BL)

Figure 8: Example of the PU modes distribution of CTUs between two successive frames using QP_{BL}=32.

Table 1: The average probability distribution of the same PU modes using (QP_{BL}, QP_{EL})=(32, 28).

Sequence	P(BL)%	P(EL)%
vidyo1	82.41	80.66
vidyo3	79.78	74.17
vidyo4	78.26	76.14
Kimonol	41.01	44.88
ParkScene	42.22	38.5
Cactus	66.13	57.24
BasketballDrill	57.13	55.67
BQMall	66.56	54.81
Average	65.44	56.5

Figure 8 shows an example of the PU modes distribution of CTUs between two successive frames in BL. From Figure 8, we can observe that the coding PU modes of neighboring CUs are correlated. Since blocks of real life video sequences are highly correlated, many PU modes in P-frame are corresponding to the same prediction modes in the tempo-spatial direction.

Table 1 shows the probability distribution of the same PU modes between neighboring and current CTU using (QP_{BL}, QP_{EL})=(32, 28) in the SHM 4.0. From Table 1, we can find that there is a high tempo-spatial correlation existing PU modes between two successive frames. Thus, when using the TSSOA to find the best candidate quadtree, the tempo-spatially neighboring PU modes can be further considered as the best prediction mode of the current CTU. After statistical analysis for the PU modes between successive frames in BL and EL, we can find that PU modes have the same probability distribution of quadtree as TSSOA. Since there is a high correlation between BL and EL, the PU modes of encoded CTU quadtree of the BL and EL frames can be utilized to speed up the process of selecting the best predicted PU modes.

The flowchart of the proposed FPUPA in BL is shown in Figure 9. The proposed FPUPA in reducing complexity for SHVC can be summarized as follows:

Step 1. Set a threshold (Thr_{QP_PU}) value for PU mode according to (QP_{BL}, QP_{EL}).

Step 2. Encode the BL of SHVC using TSSOA to find the best quadtree.

Step 3. Calculate the RDcost of PU modes in the quadtree according to the same searching order as TSSOA.

Step 4. If the RDcost is less than Thr_{QP_PU}, go to step 5. Otherwise, go to step 3.

Step 5. Use the original RDO module to prune the best CTU quadtree of the current CTU.

Step 6. Record the best CTU quadtree and corresponding parameters of BL.

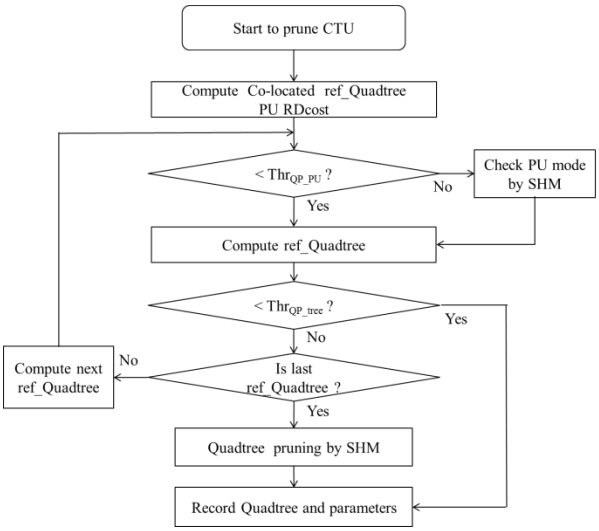


Figure 9: The flowchart of the proposed FPUPA.

3.3. Proposed FMVPA

When pruning the best CTU coding quadtree, the inter prediction module executes 7 different prediction modes to find the best mode. Especially, in the inter2N×2N 、inter2N×N 、interN×2N and interN×N prediction need perform ME modules. Since ME process full search prediction modes in each depth, this leads to requiring a very time-consuming computation in SHVC.

However, since blocks of real life video sequences are highly correlated, the MV of CU may be similar to the MVs of the co-located CU and the spatial four neighbor CUs due to tempo-spatial correlation. Figure 10 shows an example of the MV distribution of CTUs between two successive frames in BL. From Figure 10, we can observe that the MVs of neighboring CUs are correlated. In order to show the high MV correlation existing successive frames in BL, we made statistical analysis about the same MV of CU in each depth as shown in Figure 4.

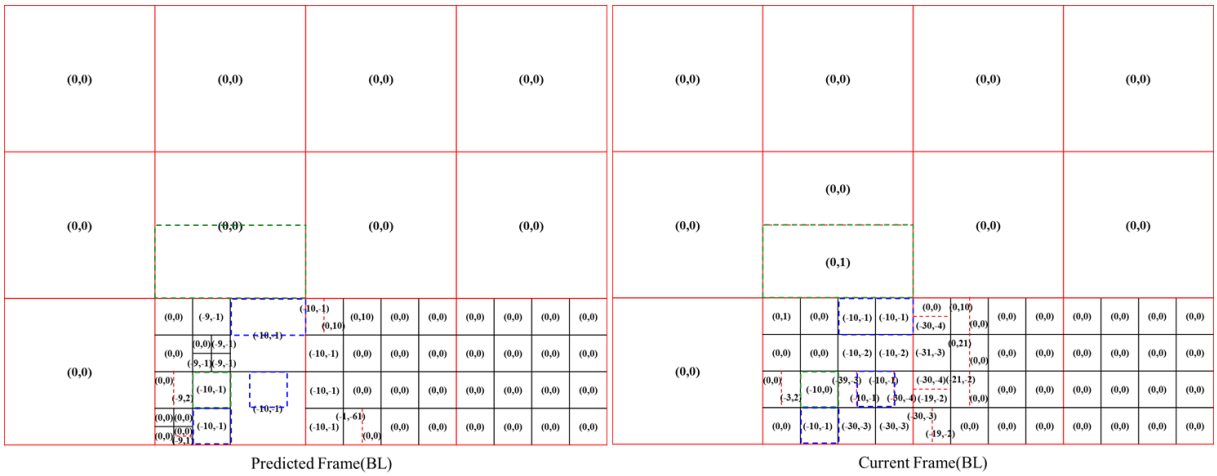


Figure 10: Example of the MV distribution for each depth of CTUs between two successive frames using QP_{BL}=32.

Table 2: The average probability distribution of the same MV using QP_{BL}=32 and Depth=1.

Sequence	$P(B_A=B_X)$	$P(B_B=B_X)$	$P(B_C=B_X)$	$P(B_D=B_X)$	$P(B_E=B_X)$
vidyo1	59.70	65.08	53.25	57.52	55.84
vidyo3	53.07	56.95	48.01	53.51	45.36
vidyo4	43.31	62.53	49.67	56.38	47.68
Kimono1	12.15	49.60	29.11	36.15	25.42
ParkScene	7.55	58.97	32.89	38.82	25.32
Cactus	55.86	60.14	49.31	55.81	50.80
BasketballDrill	66.95	61.12	54.51	61.17	59.28
BQMall	66.20	55.59	52.81	66.97	58.67
Average	45.60	58.75	46.20	53.29	46.05

Table 2 shows the probability distribution of the same MV between tempo-spatial neighboring and current CU in BL using quantization parameter QP_{BL}=32 and Depth=1 in the SHM 4.0. From Table 2, we also can find that there is a high tempo-spatial correlation of MV in each depth exists between two successive frames. Thus, when encoding the current frame in BL, the current MV in the same depth can be predicted through the MV of co-located CTU in the reference frame, and the MV of the spatial four already encoded neighboring CTUs in the current frame.

The flowchart of the proposed FMVPA in BL is shown in Figure 11. The proposed FPUPA in reducing complexity for SHVC can be summarized as follows:

Step 1. Set a threshold (Thr_{QP_MV}) value according to QP.

Step 2. Search the predict MV in BL according to search order in each depth, respectively. If the RDcost computed by priority 1 is less than Thr_{QP_MV}, go to step 6. Otherwise, go to step 3.

Step 3. Encode the BL of SHVC using FMVPA to find the best MV.

Step 4. If the RDcost less than Thr_{QP_MV}, go to step 5. Otherwise, go to step 3.

Step 5. Use the original RDO module to prune the best CTU quadtree of the current CTU.

Step 6. Record the best CTU quadtree and corresponding parameters of BL.

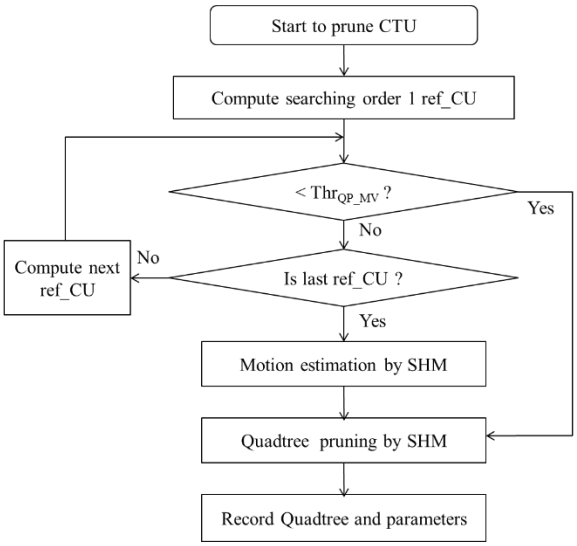


Figure 11: The flowchart of the proposed FMVPA in BL.

3.4 Proposed Fast SHVC Encoding Process

Based on the proposed FPUPA and FMVPA in BL and EL encoding procedure, respectively, we can further improve the performance of SHVC using efficient PU partition and MV predictor. Firstly, we utilize the previous TSSOA and the proposed FPUPA to speed up the encoding procedure in BL and EL. And then we employ the FCUDRD method to predict the CTU quadtree structure when TSSOA fails to work. Finally, we adopt the FMVPA to further speed up ME process. Therefore, we can implement an

early termination for split quadtree search using efficient PU mode and MV predictor methods based on a combination of previous TSSOA and FCUDRD methods. The proposed SHVC encoder does not need to go through all the modes and ME module, thus significantly reducing the computational complexity. The flowchart of the proposed complexity reduction algorithm for quality scalability in SHVC encoder is shown in Figure 12.

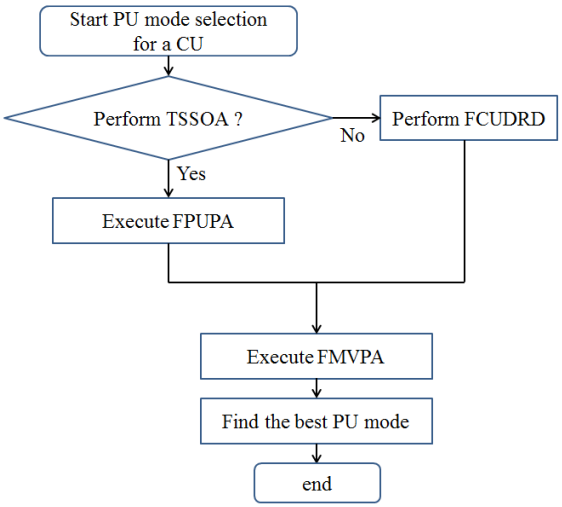


Figure 12: The flowchart of the proposed algorithm.

Table 3: Test conditions and software reference configurations.

Test sequences	Class A (2560×1600): Traffic Class B (1920×1080): Kimono, ParkScene, Cactus, BasketballDrive, BQTerrace Class C (832×480): BasketballDrill, BQMall
Total frames	33 frames
QP(BL, EL)	(22,20), (32,28), (36,32) and (40,36)
Scalability	quality scalability
Scenario	low delay (LD), random access (RA)

4. Simulation Results

For the performance evaluation, we assess the total execution time of the proposed method in comparison to those of the SHM 4.0 [14] in order to confirm the reduction in computational complexity. The system hardware is Intel (R) Core(TW) CPU i7-6700@3.40 GHz, 8.0 GB memory, and Window XP 64-bit O/S. Additional details of the encoding environment are described in Table 3.

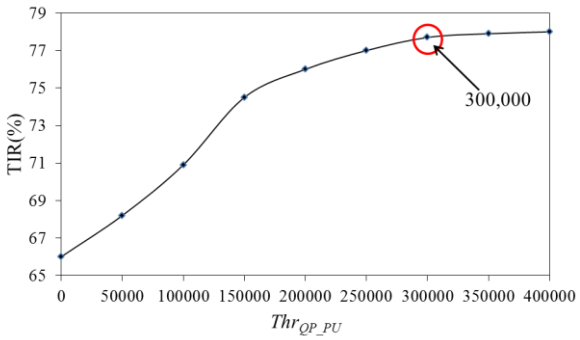
The performance of our proposed complexity reduction method is compared with those of the unmodified SHVC and previous fast encoder in terms of encoding time, bitrate and PSNRY. Note that for each video sequence the encoding time is reported for the total time of (BL+EL) in SHVC. The coding performance is evaluated based on ΔBitrate, ΔPSNRY and time improving ratio (TIR), respectively, which are defined as follows:

$$\Delta\text{Bitrate} = \frac{\text{Bitrate}_{\text{method}} - \text{Bitrate}_{\text{SHM4.0}}}{\text{Bitrate}_{\text{SHM4.0}}} \times 100\%, \quad (4)$$

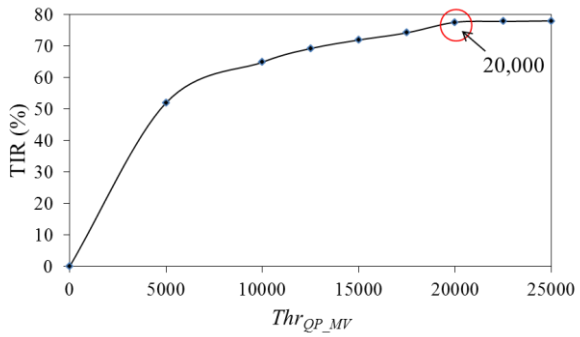
$$\Delta\text{PSNRY} = \text{PSNRY}_{\text{method}} - \text{PSNRY}_{\text{SHM4.0}}, \quad (5)$$

$$\text{TIR} = \frac{\text{TIME}_{\text{SHM4.0}} - \text{TIME}_{\text{method}}}{\text{TIME}_{\text{SHM4.0}}} \times 100\%. \quad (6)$$

Since encoding time is usually used to measure the computational complexity of the SHVC encoder, TIR measurement is adopted to assess our proposed method.



(a)



(b)

Figures 13: The average curve of TIR vs. Thr_{QP}. (a) TIR vs. Thr_{QP_PU} (b) TIR vs. Thr_{QP_MV}

Table 4: Compare FMVPA and FPOPA with SHM 4.0.

QP(32,28)	FMVPA			FPUPA		
Sequence	ΔBitrate (%)	ΔPSNRY (dB)	TIR(%)	ΔBitrate (%)	ΔPSNRY (dB)	TIR(%)
Traffic	1.75	-0.07	56.48	6.84	-0.18	70.12
Kimono	1.23	-0.02	74.24	2.41	-0.12	83.48
ParkScene	0.84	-0.03	77.84	3.55	-0.14	70.64
Cactus	1.33	-0.04	74.29	4.58	-0.12	75.42
BasketballDrive	1.64	-0.06	78.18	6.48	-0.16	78.94
BQTerrace	1.24	-0.04	60.71	4.77	-0.14	65.78
BasketballDrill	1.21	-0.05	80.48	4.89	-0.14	75.18
BQMall	0.46	-0.03	70.46	2.97	-0.13	69.81
PartyScene	0.87	-0.03	49.78	2.41	-0.14	57.48
Test	0.57	-0.03	71.43	3.24	-0.13	70.11
Average	1.11	-0.04	69.39	4.21	-0.14	71.70

The values of the threshold for FPUPA and FMVPA are two important parameters in BL and EL, which affects the coding performance of the proposed algorithm. Since the proposed algorithm is very desirable for achieving a real-time implementation of SHVC encoder, we focus on the improvement performance of encoding time. We have conducted several experiments with different thresholds values to study the effect of varying on the resulting TIR for test sequences. Figures 13(a) and 13(b) show the average curve of TIR vs. Thr_{QP} for QP_{BL}=32 , which indicates that the TIR is approximately the same for Thr_{QP_PU}≥300,000 and Thr_{QP_MV}≥20,000, respectively. From our experiment results, we find that there are high dependent relationships existing in resulting curves with various QPs. Since different QP_{BL}s could yield different average curves for TIR vs. Thr_{QP} , the thresholds are expected to be QP-dependent. Furthermore, it can be easily observed form our intensive experiments that there is a linear relationship between the threshold values and various QP_{BL} values. To mathematically model this relationship which essentially performs polynomial fitting to approximate a linear function, a linear regress model is used to derive the formula as follows [15]

Table 5: Compare FCUDRD, TSSOA and the proposed method with SHM 4.0.

QP(32,28)	Low Delay								
Sequence	FCUDRD			TSSOA			Proposed		
	ΔBitrate (%)	ΔPSNRY (dB)	TIR (%)	ΔBitrate (%)	ΔPSNRY (dB)	TIR (%)	ΔBitrate (%)	ΔPSNRY (dB)	TIR (%)
Traffic	2.56	-0.09	61.24	6.17	-0.17	65.18	7.45	-0.18	80.47
Kimono	1.11	-0.02	74.23	1.29	-0.11	78.61	2.25	-0.12	85.76
ParkScene	1.41	-0.04	65.21	3.36	-0.13	56.95	4.12	-0.14	75.42
Cactus	1.77	-0.03	55.22	5.1	-0.13	70.06	5.43	-0.14	80.18
BasketballDrive	1.02	-0.08	80.71	5.85	-0.15	73.92	7.45	-0.17	80.46
BQTerrace	-1.17	-0.03	65.46	5.41	-0.13	58.60	5.13	-0.13	76.42
BasketballDrill	2.12	-0.06	78.18	2.71	-0.14	70.77	3.48	-0.17	78.11
BQMall	1.24	-0.04	57.48	2.75	-0.12	64.49	3.41	-0.13	78.14
PartyScene	1.22	-0.02	44.72	1.98	-0.13	44.06	2.46	-0.13	69.48
Test	1.37	-0.04	55.72	2.31	-0.12	63.84	3.56	-0.13	77.28
Average	1.27	-0.05	63.82	3.69	-0.13	64.65	4.47	-0.14	78.17

$$Thr_{QP\_PU} = (\lambda_{QP} - \lambda_{32}) \times 2{,}000 + 300{,}000, \quad (7)$$

$$Thr_{QP\_MV} = (\lambda_{QP} - \lambda_{32}) \times 135 + 20{,}000, \quad (8)$$

where $\lambda = 0.4845 \times 2^{(QP-12)/3}$ is defined in SHVC specification [5].

To demonstrate the improving efficiency of our method, we test different resolution video sequences with different quantization parameter pairs under the LD and RA scenarios based on SHM 4.0. Tables 4~5 tabulate the performances obtained by testing the SHM 4.0 and the proposed method with (QP_{BL}, QP_{EL})=(32, 28) as using the LD scenario. As shown in the Table 5, the CTU processing time of proposed FPUPA and FMVPA can achieve TIR about 69.39% and 71.70% for LD on average, respectively, as compared to the SHM4.0 with insignificant loss of rate distortion (RD) performance which includes bitrate and image quality. On the other hand, Table 5 also shows the performances improvement among the FCUDRD, TSSOA and the proposed method with the same scenario. We also find that our method can further achieve an average of TIR about 14.35% and 13.52% with LD scenarios, respectively, as compared with the SHM 4.0. It is clear that the proposed fast encoding method indeed efficiently reduce the computational complexity in CU module with insignificant loss of RD performance.

In addition, we can observe that the encoding time improving is more efficient when the value of QP pairs increases, as shown in Figure 14. This is because the quantization error is too large, resulting in the lower tempo-spatial and inter-layer correlation. Furthermore, as can be seen in Tables 4~5, they also show that the TIR of CU module for Kimono and BasketballDrive sequences tested by different methods with different QP values has higher encoding reduction improvement. This is because backgrounds of these two sequences are slowly changed, and the movements are rather homogenous.

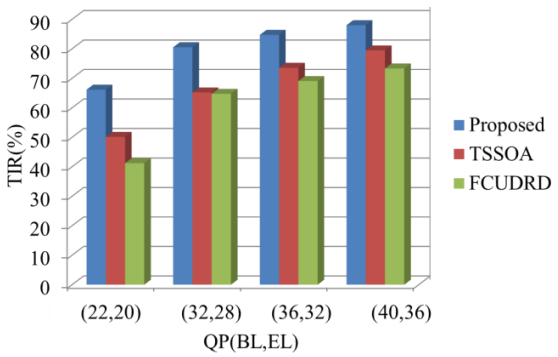


Figure 13: Average TIR using different values of QP pair.

5. Conclusion

We propose a fast PU decision algorithm for SHVC to reduce the encoding complexity. We first propose two effective encoding strategies including FPUPA and FMVPA to further improve previous FCUDRD and TSSOA methods. Simulation results show that the proposed FPUPA and FMVPA can achieve an average TIR about 69.39% and 71.70% for LD, respectively, when compared to SHM4.0. Compared with the previous algorithm implemented on SHM4.0, the proposed fast encoding method can further achieve an average TIR about 14.35% and 13.52% for LD, respectively, as compared with FCUDRD and TSSOA. It is clear that the proposed fast encoding algorithm can efficiently reduce the computational complexity of SHVC encoder with insignificant loss in terms of image quality and bitrate.

References

[1]. “Advanced video coding for generic audiovisual services”, ITU-T Rec. H.264 and ISO/IEC 14496 10, ITU-T and ISO/IEC, 2010.

[2]. H. Schwarz, D. Marpe, 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.

[3]. J. 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.

[4]. High Efficiency Video Coding. Rec. ITU-T H.265 and ISO/IEC 23008-2. Jan. 2013.

[5]. J. Boyce et al., Draft High Efficiency Video Coding (HEVC) Version 2, Combined Format Range Extensions (RExt), Scalability (SHVC), and Multi-View (MV-HEVC) Extensions, document JCTVC-R1013_v6, Sapporo, Japan, Jun./Jul. 2014.

[6]. Reference model for mixed and augmented reality defines architecture and terminology for MAR applications" (DOCX). MPEG. 2014-07-11. Retrieved 2014-07-26.

[7]. G. J. Sullivan et al., “Standardized extensions of high efficiency video coding (HEVC),” IEEE J. Selected Topics in Signal Processing, vol. 7, no. 6, pp. 1001–1016, June 2013.

[8]. C. C. Wang, Y. H. Lin and J. R. Chen, "Efficient scalable high-efficiency video coding (SHVC) using temporal-spatial searching order algorithm", The 3rd International Scientific Conference on Engineering and Applied Sciences, pp. 601-603, Okinawa, Japan, July 2015.

[9]. C. C. Wang, J. Y. Kao, "Fast encoding algorithm for H.265/HEVC based on tempo-spatial correlation", International Journal on Computer, Consumer and Control, vol. 4, no. 1, pp. 51-58, Feb. 2015

[10]. L. Shen, Z. Liu, X. Zhang, W. Zhao, Z. Zhang, “An effective CU size decision method for HEVC encoders,” IEEE Trans. Multimedia, vol. 15, no. 2, pp. 465- 470, Feb. 2013.

[11]. D. K. Kwon, M. Budagavi and M. Zhou, “Multi-loop scalable video codec based on high efficiency video coding (HEVC),” in Proc. IEEE ICASSP 2013, pp. 1749-1753, 2013.

[12]. J. Boyce, Y. Ye, J. Chen, and A. K. Ramasubramonian, “Overview of SHVC: scalable extensions of the high efficiency video coding standard”，IEEE Trans. Circuits Syst. Video Technol.,vol. 26 no. 1, pp. 20-34, Jan. 2015.

[13]. J. Chen, J. Boyce, Y. Ye, and M. M. Hannuksela, “Scalable high efficiency video coding draft 3,” in Joint Collaborative Team on Video Coding (JCT-VC) Document JCTVC-N1008, 14th Meeting: Vienna, Austria, July 25–Aug. 2 2013.

[14]. Reference software SHM4.0, https://hevc.hhi.fraunhofer.de/svn/svn_HVCSoftware/tags/SHM-4.0

[15]. David C. Lay, “Linear algebra and its applications”, University of Maryland-College Park, 5th Edition, Pearson Addison Wesley, 2016

Yuan-Shing Chang received the B.S. degree in Electrical Engineering from Tamkang University, New Taipei, Taiwan in 1994 and M.S. degree in Electrical Engineering from National Sun Yat-sen University, Kaohsiung, Taiwan in 2004, respectively. Currently, he is pursuing his PhD degree from the Department of Electronic Engineering, I-Shou University, Kaohsiung, Taiwan. His major research interests include video coding standards, IC design and test, mobile communication system and embedded system.

Ke-Nung Huang received the B.S. degree from the Department of Biomedical Engineering, Chung Yuan Christian University, Chung Li, Taiwan, in 1992 and the Ph.D. degree from the Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan, in 2003. Since 2003, he has been with the Department of Electronic Engineering, I-Shou University, Kaohsiung, Taiwan, where he is currently an Associate Professor in logic design and measurement theory. His research interests include microcomputer-based control system and intelligent instrumentation design.

Chou-Chen Wang received the B.S. and M.S. degrees in Electronic Engineering from Chung Cheng Institute of Technology (CCIT) Taoyuan, Taiwan in 1988 and 1992, respectively, and the Ph.D. degree in Electrical Engineering from National Cheng Kung University, Tainan, Taiwan in 1999. He is currently a professor with the department of Electronic Engineering at I-Shou University, Kaohsiung. His current research interests include video coding standards, image processing and compression and mobile video communication.