

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Efficient Coding Tree Unit (CTU) Decision Method for Scalable High-Efficiency Video Coding (SHVC) Encoder

Chou-Chen Wang, Yuan-Shing Chang and
Ke-Nung Huang

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/64847>

Abstract

High-efficiency video coding (HEVC or H.265) is the latest video compression standard developed by the joint collaborative team on video coding (JCT-VC), finalized in 2013. HEVC can achieve an average bit rate decrease of 50% in comparison with H.264/AVC while still maintaining video quality. To upgrade the HEVC used in heterogeneous access networks, the JCT-VC has been approved scalable extension of HEVC (SHVC) in July 2014. The SHVC can achieve the highest coding efficiency but requires a very high computational complexity such that its real-time application is limited. To reduce the encoding complexity of SHVC, in this chapter, we employ the temporal-spatial and inter-layer correlations between base layer (BL) and enhancement layer (EL) to predict the best quadtree of coding tree unit (CTU) for quality SHVC. Due to exist a high correlation between layers, we utilize the coded information from the CTU quadtree in BL, including inter-layer intra/residual prediction and inter-layer motion parameter prediction, to predict the CTU quadtree in EL. Therefore, we develop an efficient CTU decision method by combing temporal-spatial searching order algorithm (TSSOA) in BL and a fast inter-layer searching algorithm (FILSA) in EL to speed up the encoding process of SHVC. The simulation results show that the proposed efficient CTU decision method can achieve an average time improving ratio (TIR) about 52–78% and 47–69% for low delay (LD) and random access (RA) configurations, respectively. It is clear that the proposed method can efficiently reduce the computational complexity of SHVC encoder with negligible loss of coding efficiency with various types of video sequences.

Keywords: video standards, video compression, high-efficiency video coding (HEVC), scalable high-efficiency video coding (SHVC), temporal-spatial correlation

1. Introduction

With the advanced researches of electronic technology, the panels of $4K \times 2K$ (or $8K \times 4K$) high-resolution have become the main specification of large size digital TV in future. On the other hand, with rapid development of Internet and mobile devices, more and more people browse high-quality video content by smart phone or laptop, which greatly enrich people's lives. However, the currently state-of-the-art video coding standard H.264/advanced video coding (AVC) is difficult to support the video applications of high definition (HD) and ultrahigh definition (UHD) resolution. Therefore, a new video coding standard called high-efficiency video coding (HEVC) has been standardized by the Joint Collaborative Team on Video Coding (JCT-VC) jointly established by the ITU-T and ISO/IEC to satisfy the UHD requirement in January 2013, and the first edition of HEVC was approved as ITU-T H.265 and ISO/IEC 23008-2 by JCT-VC [1]. The goal of H.265/HEVC is to achieve roughly 50% bitrate reduction over H.264/AVC while still maintaining video quality [2–6]. The HEVC adopts the quadtree-structured coding tree unit (CTU), and each CTU allows recursive splitting into four equal coding units (CUs) where each CU can have the prediction unit (PU) and transform unit (TU). The HEVC can achieve the highest coding efficiency but requires a very high computational complexity so that it is difficult to be used for real-time applications. On the other hand, traditional client-server video streaming has been unable to satisfy people's ever-growing demands for video applications using heterogeneous devices and networks including the Internet and mobile network nowadays. To overcome this problem, scalable video coding (SVC) can provide an attractive solution using a single bitstream to simultaneously serve various devices with different display resolution and image fidelities. Therefore, to upgrade the HEVC further used in heterogeneous access networks, the JVT-CT develops a scalable extension of HEVC (SHVC) and is finalized in July 2014 [7, 8]. SHVC mainly includes spatial scalability, temporal scalability and quality/signal-to-noise ratio (SNR) scalability. Based on the HEVC, the SHVC scheme supports multi-loop solutions by enabling different inter-layer prediction (ILP) mechanisms [9–12]. Although the SHVC can achieve the highest coding efficiency, it requires a higher computational complexity than HEVC standard. As a result, the very high encoding complexity of SHVC has become a main obstruction for the real-time services.

To reduce the computational complexity of the SHVC encoder, there are many fast methods with negligible losses of image quality have been proposed recently [13–17]. Tohidypour et al. reduced the coding complexity of spatial or SNR/quality/fidelity scalability in SHVC using an adaptive range search method according to statistical properties [13–16]. Bailleul et al. speeded up the encoding process in enhancement layer (EL) using a fast mode decision for SNR scalability in SHVC [16]. Qingyangl et al. also proposed a fast encoding method using maximum encoding depth based on the correlation between the base layer (BL) and EL for SNR scalability in SHVC encoder and greatly reduce encoding time in BL and EL, respectively [17]. Although these methods can reduce the complexity of the encoding process for SHVC in different level with different complexity calculation method, their methods are used only in the correlation of CU depth and modes existing in BL and EL. So, the complexity of the whole encoder still has the room to be further reduced.

To overcome the drawback of huge encoding computation in SHVC, we firstly propose a temporal-spatial searching order algorithm (TSSOA) to speed up the encoding procedure in BL. Second, we develop a fast inter-layer searching algorithm (FILSA) in EL to predict the CTU quadtree structure. There are five encoded temporal-spatial causal neighbouring CTUs are chosen to be predicted by the TSSOA in BL, which shows the searching priority order according to the correlation values which are determined by values of statistic. Due to the less data information and high correlation existing in residual image in EL, thus only three encoded inter-layer causal neighbouring CTUs are chosen to be predicted by the FILSA in EL.

2. SHVC background

HEVC can greatly improve coding efficiency by adopting hierarchical structures of CU, PU and TU. The CU depths can be split by coding quadtree structure of four level, and the CU size can vary from 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 (CU size), 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) at maximally three level depths which vary from 32×32 to 4×4 pixels. The relationship of hierarchical CU, PU and TU coding structure of HEVC is shown in **Figure 1** [2–6].

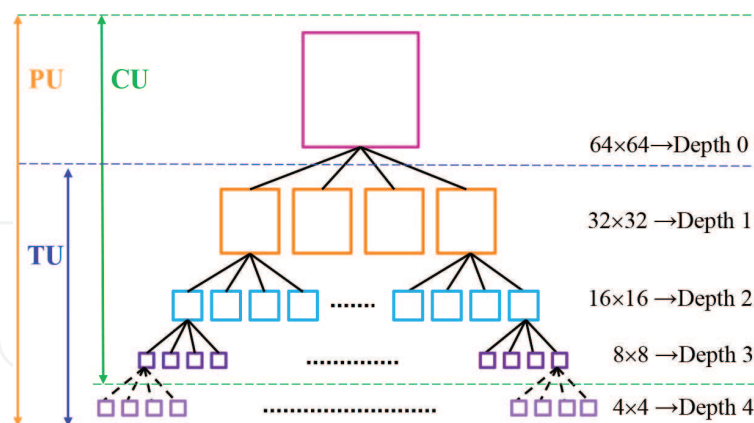


Figure 1. The relationship of hierarchical CU, PU and TU coding structure of HEVC [6].

In general, intra-coded CUs have only two PU partition types including $2N \times 2N$ and $N \times N$ but inter-coded CUs have eight PU types including symmetric blocks ($2N \times 2N$, $2N \times N$, $N \times 2N$, $N \times N$) and asymmetric blocks ($2N \times nU$, $2N \times nD$, $nL \times 2N$, $nR \times 2N$) [4]. When only using symmetric PU blocks, H.265/HEVC encoder tests seven different partition sizes including SKIP, inter $2N \times 2N$, inter $2N \times N$, inter $N \times 2N$, inter $N \times N$, intra $2N \times 2N$ and intra $N \times N$ for inter-slice as shown in **Figure 2**. 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. Since all the PUs and available TUs have to be exhaustively searched by rate-distortion optimization (RDO) process for an LCU, H.265/HEVC dramatically increased computational complexity compared with H.264/AVC [4, 5]. The optimization of the block mode decision procedure will result in the high computational complexity and limit the use of HEVC encoders in real-time applications. Since the coding procedure for HEVC is very complex, the coding procedure for SHVC is even more complex due to an extension of HEVC.

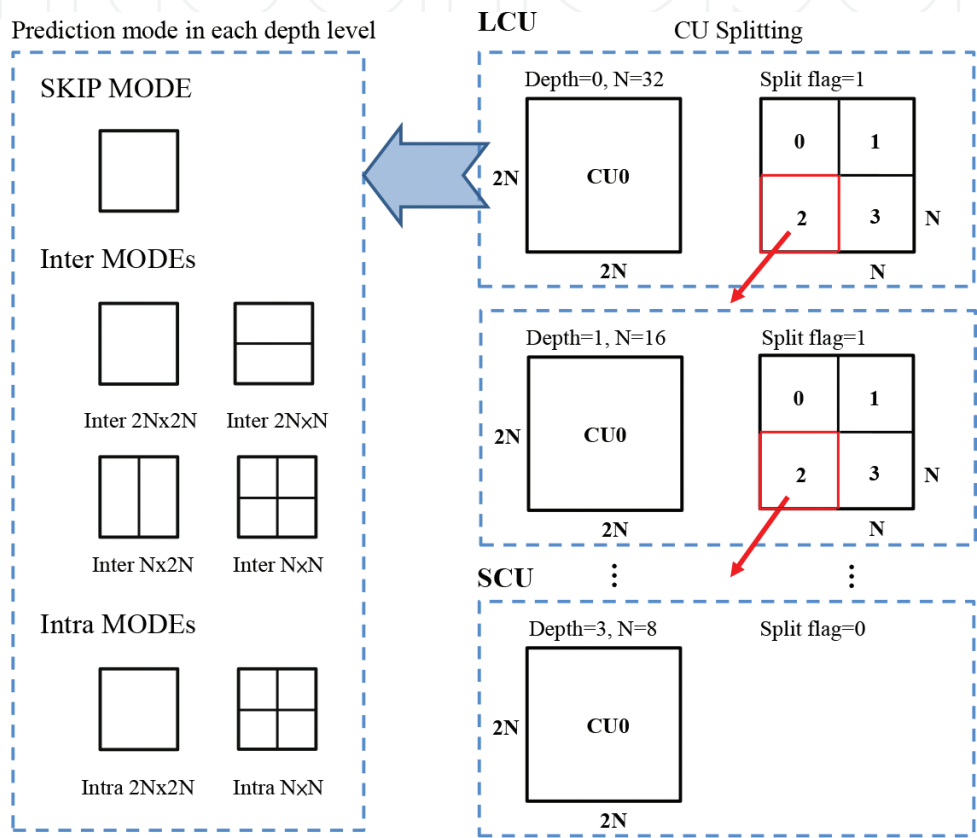


Figure 2. The architecture of quadtree structured CUs and PU partitioning [6].

Based on the HEVC, the SHVC scheme supports both single-loop and multi-loop solutions by enabling different inter-layer prediction (ILP) mechanisms [18, 19]. A typical architecture of two-layer SHVC encoder is shown in **Figure 3**. However, SHVC encoder allows one BL and more than one EL. **Figure 3** illustrates how the decoded BL picture is used for prediction in EL coding in a two-layer SHVC encoder. The input video of BL can be encoded or decoded with HEVC coding tools. The decoded picture of BL is processed by the ILP module before being sent to the decoded picture buffer (DPB) of EL. For the EL, the BL decoded picture which obtained by ILP is called as the inter-layer reference (ILR) picture. The ILP module performs inter-layer intra/residual prediction and inter-layer motion parameter prediction by upsampling calculations. Furthermore, the discrete cosine transform/quantization (DCT/Q) and inverse DCT/inverse quantization (IDCT/IQ) modules are further applied to inter-layer

prediction residues for better energy compaction. The parameters used for such EL, shown as ILP information in **Figure 3**, are multiplexed together with BL and EL bitstreams to form an SHVC bitstream. 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) scalability. From the Reference [18], we can find that the encoding complexity of HEVC is higher than that of H.264/AVC encoder. Therefore, the computational burden of SHVC encoder is expected to be several times more than HEVC encoder. Nowadays, it is an important topic to study how to reduce the computational complexity of SHVC to achieve real-time applications.

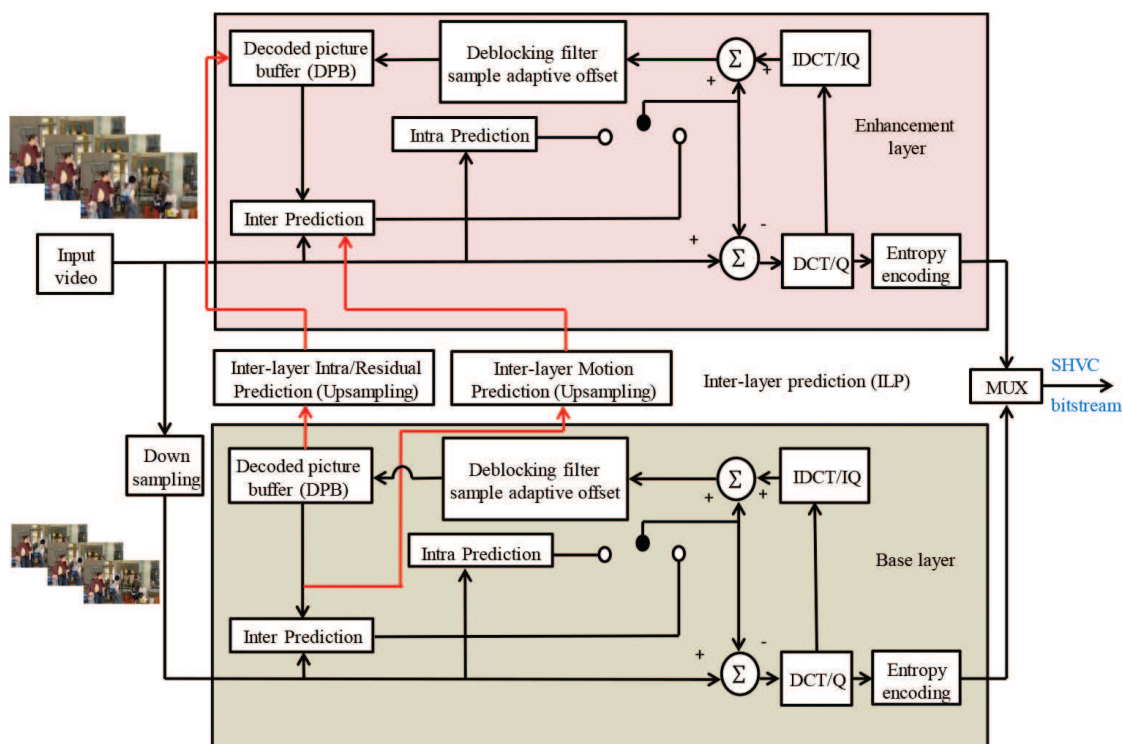


Figure 3. A typical architecture of two-layer SHVC encoder.

3. Proposed CTU decision method

Each layer encoding process in SHVC can be considered similar with HEVC, except for the enhancement layers using inter-layer prediction techniques. However, the computational complexity of the HEVC encoder increases dramatically due to its recursive quadtree representation to find the best CTU partition. Therefore, we can know that the computational complexity of SHVC encoder is more than HEVC encoder. Thus, we utilize the temporal-spatial correlation prediction in BL based on HEVC and inter-layer correlation prediction in EL to develop an efficient CTU decision method to speed up SHVC encoding process.

3.1. Temporal-spatial correlation in BL

As the frame rate highly increasing, the successive two frames have a stronger temporal-spatial correlation. **Figure 4** shows two certain frames of the test sequence encoded using low-delay (LD) configuration in BL by the SHVC reference software (SHM 6.0) [21]. As shown in **Figure 4**, the quadtree structures of the CTU in the current frame, for example **Figure 4** (A_0) and $4(A'_0)$, are the same as or similar to the split quadtree structures of the temporally co-located coded CTUs of the previous frame shown in **Figure 4**. On the other hand, there are also the same as or similar to the split structures of the spatial four neighbour CTUs in the current frame, for example **Figure 4**(B–E). **Figure 4** shows the corresponding five causal encoded neighbouring CTUs (A–E) of the current CTU(X) in the temporal-spatial direction.

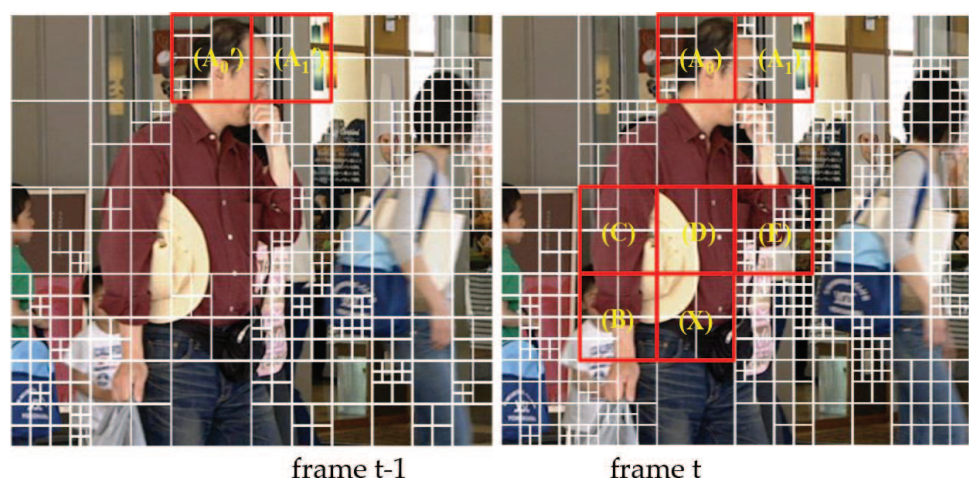


Figure 4. Examples of the quadtree structures of CTU between two successive frames in BL.

As observed and described above, there is always a high correlation existing encoded frames in BL. In order to show the temporal-spatial correlation existing successive frames in BL, we made statistical analysis about the optimal quadtree structure of encoded CTU in BLs. **Figure 5** shows the corresponding five causal encoded neighbouring CTUs ($B_A \sim B_E$) of the current CTU(X) in the temporal-spatial direction in BL, respectively.

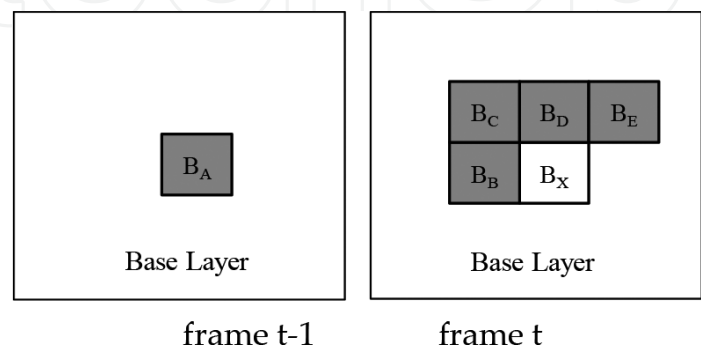


Figure 5. Corresponding five causal encoded neighbouring CTUs of the current CTU(X) in BL.

Table 1 shows the probability distribution of the same CTU quadtree between temporal-spatial neighbouring and current CTU in BL using quantization parameter $QP_{BL} = 32$ and 100 frames in the SHM 6.0. From **Table 1**, we can find that there is a high temporal-spatial correlation of quadtree exists between two successive frames. Thus, when encoding the current frame in BL, the current CTU can be predicted through the split quadtree structure of the co-located CTU in the reference frame and the split quadtree structure of the spatial four already encoded neighbouring CTUs in the current frame.

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	77.15	73.03	56.01	61.39	55.63
Vidyo3	76.07	70.09	55.59	61.59	55.02
Vidyo4	72.44	67.34	53.06	58.76	52.82
Kimono	33.71	30.55	22.49	27.51	22.13
ParkScene	35.80	36.81	29.67	34.60	29.10
Basketball	46.01	49.42	39.68	43.38	40.55
Cactus	52.69	48.57	39.69	45.35	40.43
BQTerrace	45.85	45.57	35.42	40.45	35.79
Average	54.97	52.92	41.45	46.63	41.43

Table 1. The probability distribution of the same CTU quadtree between temporal-spatial neighbouring and current CTU using $QP_{BL} = 32$.

3.2. Inter-layer correlation between BL and EL

As described in Section 2, there is always a strong inter-layer correlation when adopting layer-based encoding structure. In the same situation for SHVC, we can expect that there exists a high inter-layer correlation between BL and EL when using quality scalability configuration, which BL and ELs have the same resolution with different QP. In order to find the inter-layer correlation between BL and EL, we statistically analyse the split quadtree structures of encoded CTU in BL and EL with different video sequences. In this experiment, we find that there exists a high inter-layer correlation between BL and EL. The results we got are similar to temporal-spatial correlation in BL. **Figure 6** shows the examples of the quadtree structures of CTU between BL and EL in the same frame. As shown in **Figure 6**, the quadtree structures of the CTU in the BL, for example **Figure 6**(X_0) and (X_1), are the same as or similar to the split quadtree structures of the corresponding co-located coded CTUs, **Figure 6** (X'_0) and (X'_1) in the EL. **Figure 7** shows the corresponding six causal encoded neighbouring CTUs ($E'_X, E_B \sim E_E$) and CTU(B_A) of the current CTU(X) in the temporal-spatial direction in EL and in the inter-layer between EL and BL, respectively.

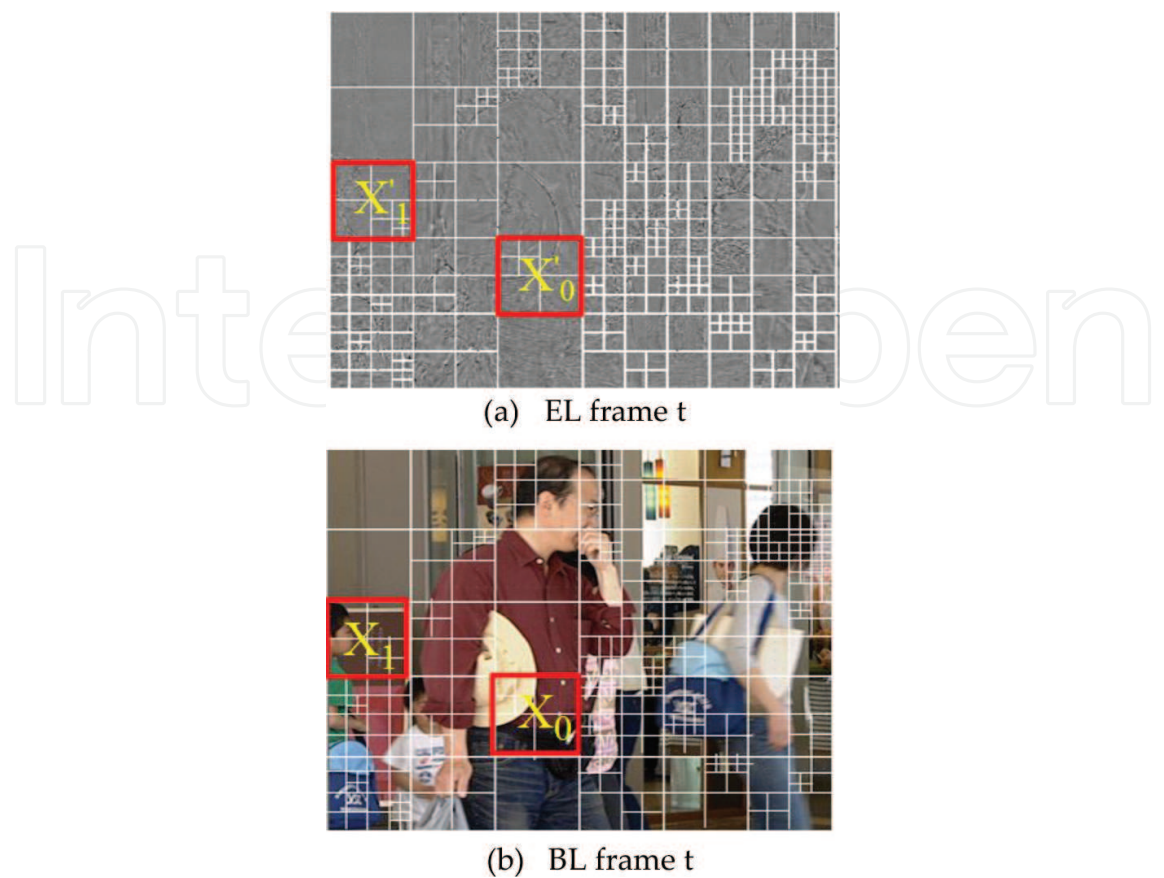


Figure 6. Examples of the quadtree structures of CTU between BL and EL in the same frame.

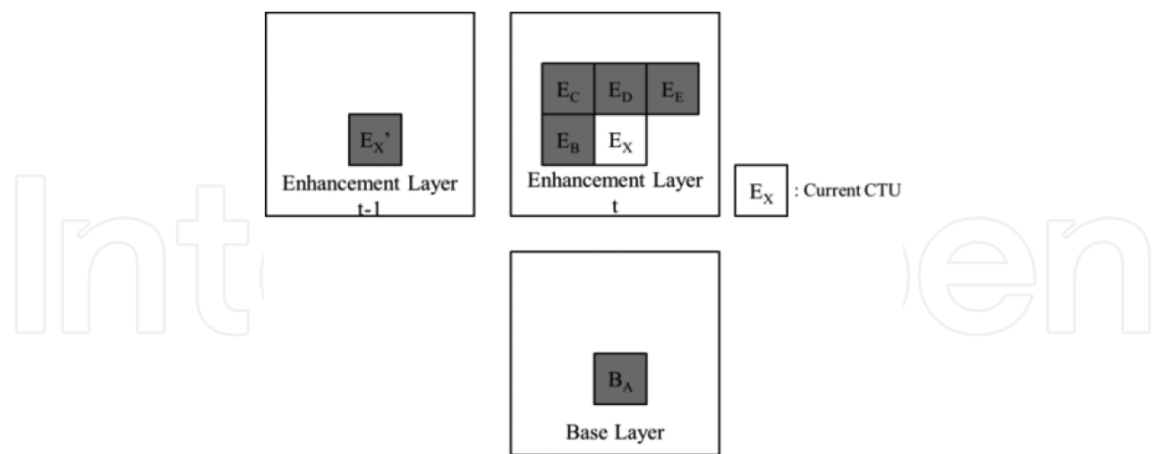


Figure 7. Example of the corresponding six causal encoded neighbouring CTU between BL and EL.

In the same procedure as BL, to show the inter-layer correlation existing in the same frame between BL and EL, we made statistical analysis about the optimal quadtree structure of encoded CTU between EL and BL. In addition, we also made statistical analysis for the temporal-spatial correlation existing successive frames in EL at the same time. **Table 2** shows

the probability distribution of the same CTU quadtree between BL and EL using $QP_{(BL, EL)} = QP_{(32, 28)}$ in the SHM 6.0. In the same situation, we can find that there is a high inter-layer correlation exists between BL and EL. Since there is a high correlation between BL and EL, the encoded CTU quadtree of the BL frames can be utilized to speed up the process of selecting the best predicted CTU quadtree for the corresponding EL frames [20]. Besides, the already encoded neighbouring CTUs in the EL are valuable for predicting the quadtree of the current CTU. Therefore, the temporal-spatial neighbouring encoded CTUs in the EL and the inter-layer corresponding encoded CTU in the BL are used to predict the current CTU in EL. From **Table 2**, we can find that there is a higher inter-layer correlation exists between BL and EL except for temporal-spatial correlation in EL. In addition, we also find that the probability distributions of $CTU(E_C)$, $CTU(E_E)$ and $CTU(E'_X)$ are almost the same and less than the others. For simplicity, when encoding the current frame in EL, the current $CTU(X)$ can be interlayer predicted by the split quadtree structure of CTU in BL and then predicted through the split quadtree structure of the two split structure of the spatial already encoded neighbouring CTUs in EL.

Sequence	$P(B_A = E_X)\%$	$P(E_B = E_X)\%$	$P(E_C = E_X)\%$	$P(E_D = E_X)\%$	$P(E_E = E_X)\%$	$P(E_X = E_X)\%$
Vidyo1	74.88	60.41	44.10	51.54	45.23	48.37
Vidyo3	74.03	61.53	44.50	51.74	44.23	46.84
Vidyo4	75.59	63.57	45.89	51.31	45.06	49.19
Kimono	32.52	32.18	24.06	27.95	23.38	26.58
ParkScene	41.45	27.53	21.56	26.35	19.77	23.67
Basketball	53.44	47.70	33.22	37.08	32.24	37.73
Cactus	54.31	43.69	35.40	41.09	35.93	36.51
BQTerrace	39.30	27.87	20.08	23.27	21.00	22.71
Average	55.69	45.56	33.60	38.79	33.35	36.45

Table 2. The probability distribution of the same CTU quadtree between inter-layer neighbouring and current CTU using $QP_{(BL, EL)} = QP_{(32, 28)}$.

3.3. Fast SHVC encoder using efficient CTU decision

3.3.1 Temporal-spatial searching order algorithm (TSSOA)

To speed up the encoding process of SHVC in BL, we propose a temporal-spatial searching order algorithm (TSSOA) which utilizes the characteristics of natural video sequence existing strongly temporal and spatial correlation. In this work, the five causal neighbouring encoded split quadtree structures of CTUs shown in **Figure 5**, on temporal-spatial direction, are firstly chosen as candidates for the current CTU encoding in BL. **Figure 8** shows the search priority order according to the sorted correlation values determined by experiments from **Table 1**. Block 1 represents the temporal neighbour, and blocks 2–5 denote spatial neighbours 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, we check compute the RD cost using the predicted split structure. After the candidate split structure (one of blocks 1–5) is found, we check whether it is good enough for the current CTU by comparing its RD cost with a threshold (Thr). If it is less than the threshold, the candidate is good enough for the current CTU. Otherwise, it implies that the temporal-spatial correlation is low and a full recursive process is needed to find the optimal split quadtree structure of the current CTU.

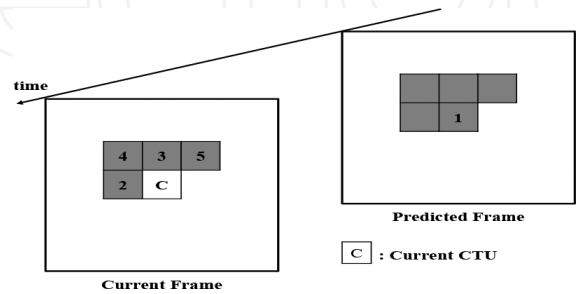


Figure 8. The search priority order.

The flow chart of the proposed TSSOA is shown in Figure 9.

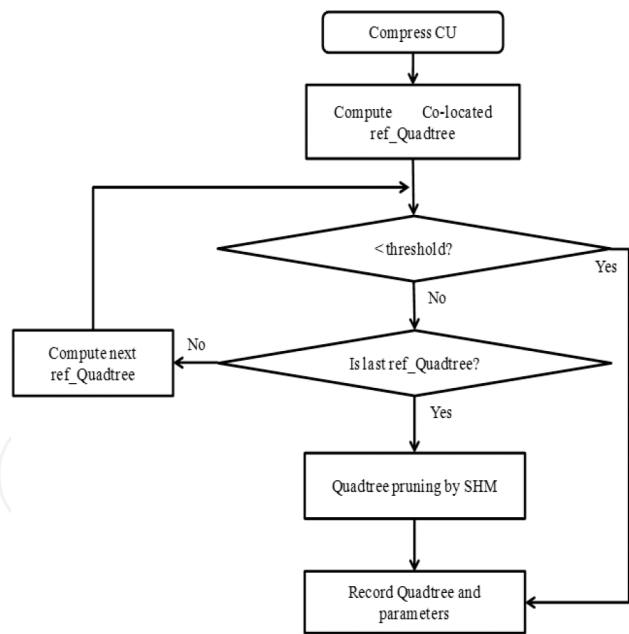


Figure 9. The flow chart of the proposed TSSOA.

The proposed TSSOA in the fast encoding for SHVC can be summarized as follows:

Step 1. Set a threshold (Thr) value according to QP.

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

- Step 3.** If it is last neighbouring CTU, go to step 5. Otherwise, go to step 4.
- Step 4.** Compute RDcost of next neighbouring CTU in the priority order (2–5), if the RDcost less than Thr, go to step 6. 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.

3.3.2. Fast inter-layer searching algorithm (FILSA)

For fast EL encoding, we use the fast inter-layer searching algorithm (FILSA) between BL and EL to predict the split quadtree structure of CTU for the current CTU in EL. Due to the less data information and very high correlation existing in residual image in EL, thus only three causal neighbouring split quadtree structure of CTUs shown in **Figure 10** are chosen as the candidates. This is because we find that there is a highest inter-layer correlation existing CTU(B_A), CTU(E_B) and CTU(E_D). In other words, we eliminate CTU(E_C), CTU(E_E) and CTU(E_X) as candidates since their probability distributions are almost the same and less than the others. Therefore, when encoding the current frame in EL by FILSA, the current CTU(X) can be interlayer predicted by the split quadtree structure of CTU in BL and then predicted through the two split quadtree structure of the spatial already encoded neighbouring CTUs in EL. The FILSA determines that split quadtree structure of CTUs is the best candidate for the current CTU in EL, and it computes the RD costs from the predicted split quadtree of CTUs and selects the minimum RD cost as the best split quadtree of CTU(E_X). From our experiments, we can verify the encoding performance with negligible decrease when only utilizes three candidates as shown in **Figure 10**.

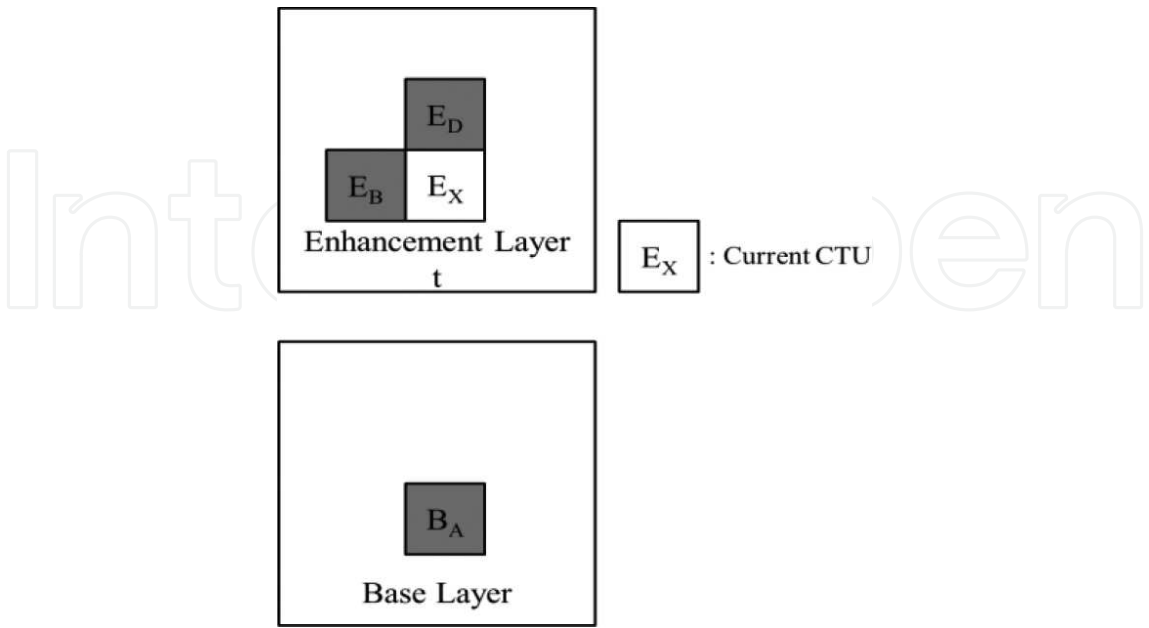


Figure 10. Three causal neighbouring split quadtree structure of CTUs as candidates.

3.3.3. Fast SHVC encoder

Based on the proposed TSSOA and FILSA in BL and EL encoding procedure, respectively, we can develop a fast SHVC encoder using efficient CTU decisions. First, we utilize the TSSOA to speed up the encoding procedure in BL. Second, we employ the FILSA to predict the CTU quadtree structure in ELs. Therefore, we can implement an early termination (ET) for split quadtree search using an efficient CTU decision method based on combining the proposed TSSOA and FILSA. The proposed SHVC encoder does not need to go through all the modes, thus significantly reducing the computational complexity. The flow chart of the proposed fast SHVC encoder is shown in **Figure 11**.

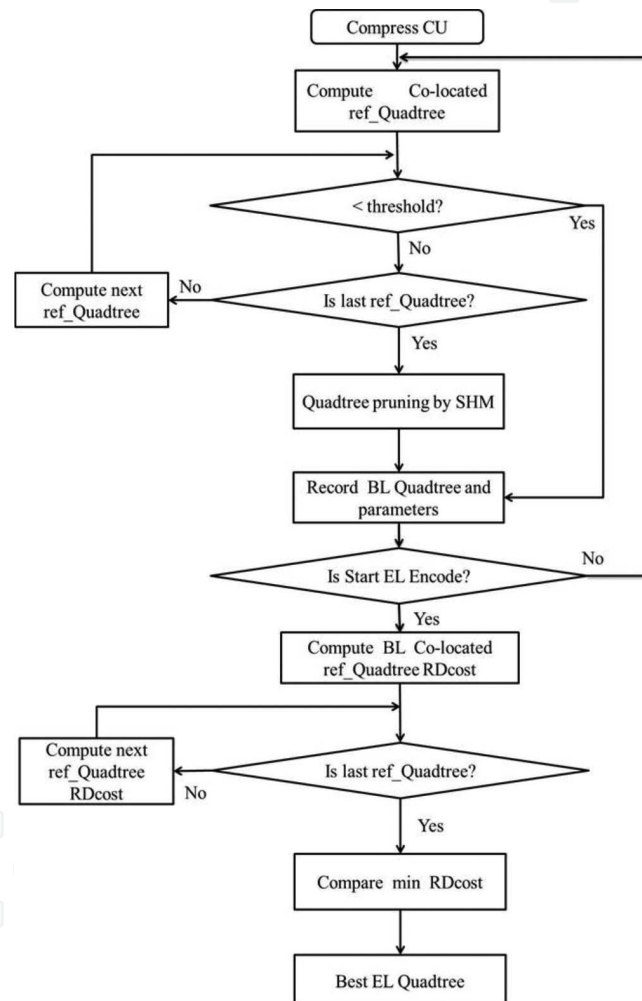


Figure 11. The flow chart of the proposed fast SHVC encoder.

4. Simulation results and discussion

For the performance evaluation, we assess the total execution time of the proposed method in comparison with those of the SHM 6.0 [21] in order to confirm the reduction in computational

complexity. The system hardware is Intel (R) Core(TW) CPU i5-3350P @ 3.10 GHz, 8.0 GB memory, and Window XP 64-bit O/S. Additional details of the encoding environment are described in **Table 3**.

Test sequences	Class A (2560 × 1600): Traffic Class B (1920 × 1080): Kimono, ParkScene, Cactus, BasketballDrive and BQTerrace Class C (832 × 480): BasketballDrill, BQMall, PartyScene
Total frames	100 frames
Quantization parameter $QP_{(BL, EL)}$	(22, 20), (32, 28), (36, 32) and (40, 36)
Software	SHM 6.0
Scenario	Low delay (LD), random access (RA)

Table 3. Test conditions and software reference configurations.

The performance of our proposed complexity reduction method is compared with that of the unmodified SHVC encoder in terms of encoding time, impact on bitrate and peak signal-to-noise ratio of Y component (PSNRY). Note that for each video sequence, the encoding time is reported for the total time (BL + EL). The coding performance is evaluated based on Δ Bitrate, Δ PSNRY and time improving ratio (TIR), respectively, which are defined in Eqs. (1–3) and described as follows:

$$\Delta \text{Bitrate} = \frac{\text{Bitrate}_{\text{proposed}} - \text{Bitrate}_{\text{SHM6.0}}}{\text{Bitrate}_{\text{SHM6.0}}} \times 100\%, \quad (1)$$

where the ratio of encoding bitrate reduction is represented by Δ Bitrate, and $\text{Bitrate}_{\text{proposed}}$ and $\text{Bitrate}_{\text{SHM 6.0}}$ represent the encoding bitrate of the proposed method and the conventional method based on the SHM 6.0 reference software, respectively.

$$\Delta \text{PSNRY} = \text{PSNRY}_{\text{proposed}} - \text{PSNRY}_{\text{HM6.0}} \quad (2)$$

where Δ PSNRY is the ratio of encoding quality reduction, and $\text{PSNRY}_{\text{proposed}}$ and $\text{PSNRY}_{\text{HM 6.0}}$ represent the proposed method and the SHM 6.0, respectively.

$$\text{TIR} = \frac{\text{TIME}_{\text{proposed}} - \text{TIME}_{\text{SHM6.0}}}{\text{TIME}_{\text{SHM6.0}}} \times 100\% \quad (3)$$

where TIR is the ratio of encoding time reduction, and $\text{TIME}_{\text{proposed}}$ and $\text{TIME}_{\text{SHM 6.0}}$ represent the proposed method and the SHM 6.0, respectively. Encoding time is usually used to measure

the computational complexity of the SHVC encoder, and thus, a TIR measurement is adopted to assess our proposed fast method.

The value of the threshold (Thr) for TSSOA is an important parameter in BL encoding, which affects the coding performance of the proposed algorithm. A lower value of means that more RDOs are performed to prune the best CTU quadtree, and thus, more time is spent to encode them and a closer quality to that of SHM 6.0 will be obtained. However, since the proposed fast 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 values of Thr to study the effect of varying t on the resulting TIR for test sequences shown in **Table 3**. **Figure 12** shows the average curve of TIR vs. Thr_{QP} for QP_{BL} = 32 which indicates that the TIR is approximately the same for all $\geq 350,000$. From our experiment results, we find that there are high dependent relationships existing 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 from our intensive experiments that there is a linear relationship between the threshold values and the 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 [20]

$$\text{Thr}_{\text{QP}} = (\lambda_{\text{QP}} - \lambda_{32}) \times 4350 + 350,000 \quad (4)$$

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

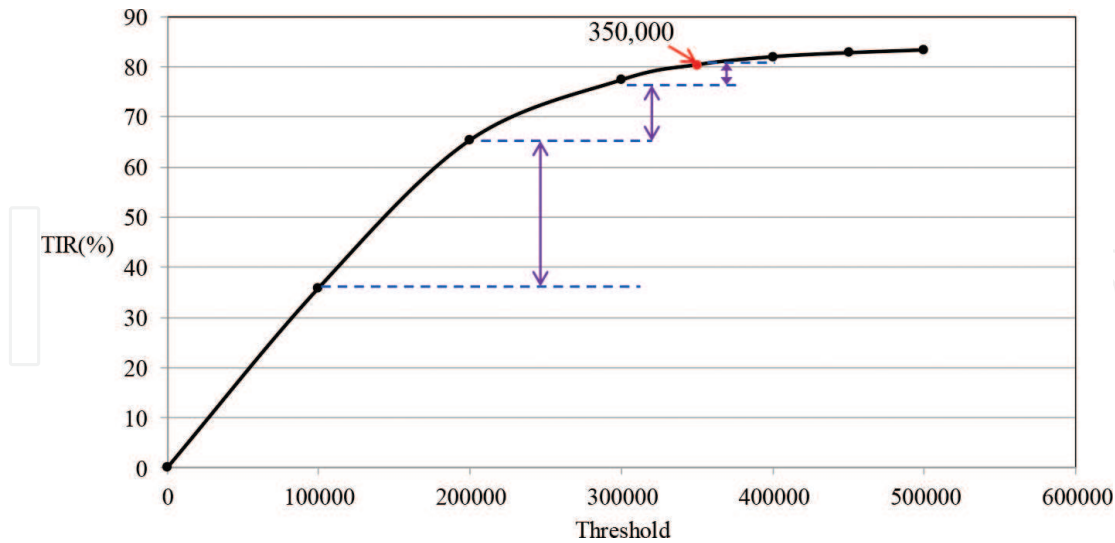


Figure 12. The average curve of TIR vs. Thr_{QP} for QP_{BL} = 32.

Tables 4–7 tabulate the performances obtained by testing the SHM 6.0 and the proposed method with different quantization parameter pairs when uses the random access (RA) and LD scenarios, separately. The simulation results show that the proposed algorithm can reduce

the computational complexity of CTU quadtree pruning of SHVC about 34~71% when compared to SHM 6.0. From **Tables 4–7**, we find that the proposed fast SHVC encoder can further achieve an average TIR about 47~78%. In addition, we can observe that the encoding time improving is more efficient when the value of QP pairs increases. This is because the quantization error is too large that results in the lower temporal-spatial and inter-layer correlation. Furthermore, as can be seen in **Tables 4–7**, 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.

QP _(22, 20)	RA			LD		
Sequence	Proposed/SHM 6.0			Proposed/SHM 6.0		
	ΔBitrate (%)	ΔPSNRY (dB)	TIR (%)	ΔBitrate (%)	ΔPSNRY (dB)	TIR (%)
Traffic	2.95	−0.17	−29.13	2.19	−0.13	−36.13
Kimono	0.90	−0.12	−62.27	0.76	−0.09	−64.45
ParkScene	1.04	−0.14	−42.41	0.92	−0.11	−46.62
Cactus	1.49	−0.15	−45.33	1.27	−0.09	−45.10
BasketballDrive	2.78	−0.18	−46.81	2.48	−0.15	−63.06
BQTerrace	2.09	−0.11	−42.24	1.86	−0.10	−50.62
BasketballDrill	0.34	−0.13	−50.64	0.28	−0.10	−55.63
BQMall	0.81	−0.11	−50.37	0.74	−0.10	−51.39
PartyScene	0.47	−0.13	−50.70	0.32	−0.09	−52.17
Average	1.93	−0.14	−46.66	1.21	−0.11	−51.68

Table 4. Comparison of the proposed method with SHM 6.0 using QP_(22, 20).

QP _(32, 28)	RA			LD		
Sequence	Proposed/ SHM 6.0			Proposed/ SHM 6.0		
	ΔBitrate (%)	ΔPSNRY (dB)	TIR (%)	ΔBitrate (%)	ΔPSNRY (dB)	TIR (%)
Traffic	6.64	−0.21	−43.06	6.18	−0.17	−65.18
Kimono	1.87	−0.13	−71.26	1.38	−0.11	−78.61
ParkScene	3.62	−0.15	−62.06	3.34	−0.13	−56.95
Cactus	5.69	−0.17	−57.69	5.07	−0.15	−70.06
BQTerrace	5.96	−0.16	−56.64	5.46	−0.13	−58.60
Basketball	3.10	−0.13	−60.03	2.82	−0.11	−70.77
BQMall	2.81	−0.14	−58.96	2.74	−0.12	−64.49
PartyScene	2.21	−0.15	−58.52	1.97	−0.13	−44.06
Average	4.28	−0.16	−59.96	3.88	−0.13	−65.18

Table 5. Comparison of the proposed method with SHM 6.0 using QP_(32, 28).

QP _(36,32)	RA			LD		
Sequence	Proposed/SHM 6.0			Proposed/SHM 6.0		
	Δ Bitrate (%)	Δ PSNRY (dB)	TIR (%)	Δ Bitrate (%)	Δ PSNRY (dB)	TIR (%)
Traffic	5.98	-0.17	-52.32	5.71	-0.14	-65.77
Kimono	3.84	-0.11	-78.23	3.46	-0.09	-83.23
ParkScene	4.27	-0.12	-66.29	3.93	-0.10	-75.21
Cactus	5.14	-0.14	-65.36	4.83	-0.11	-66.22
BasketballDrive	7.53	-0.16	-76.68	7.21	-0.13	-81.00
BQTerrace	5.89	-0.11	-63.88	5.27	-0.09	-73.99
BasketballDrill	2.70	-0.13	-65.70	2.38	-0.13	-82.49
BQMall	3.87	-0.14	-58.77	3.40	-0.11	-66.90
PartyScene	4.02	-0.12	-61.34	3.77	-0.10	-67.13
Average	5.14	-0.13	-65.73	4.44	-0.11	-73.55

Table 6. Comparison of the proposed method with SHM 6.0 using QP_(36,32).

QP _(40,36)	RA			LD		
Sequence	Proposed/SHM 6.0			Proposed/SHM 6.0		
	Δ Bitrate (%)	Δ PSNRY (dB)	TIR (%)	Δ Bitrate (%)	Δ PSNRY (dB)	TIR (%)
Traffic	8.04	-0.18	-63.00	7.13	-0.14	-77.74
Kimono	5.06	-0.14	-80.04	4.51	-0.12	-85.82
ParkScene	5.12	-0.13	-71.59	4.37	-0.11	-79.71
Cactus	7.33	-0.15	-72.93	5.88	-0.12	-78.36
BasketballDrive	8.26	-0.16	-75.92	7.43	-0.14	-82.27
BQTerrace	7.21	-0.15	-65.76	6.14	-0.15	-75.33
BasketballDrill	4.15	-0.14	-68.43	3.72	-0.13	-76.01
BQMall	4.86	-0.15	-60.40	4.06	-0.13	-69.86
PartyScene	4.22	-0.13	-66.15	3.49	-0.11	-74.52
Average	6.03	-0.15	-69.36	5.19	-0.13	-77.74

Table 7. Comparison of the proposed method with SHM 6.0 using QP_(40,36).

In summary, the results show the superiority of our proposed fast efficient CTU decision including TSSOA and FILSA over the state-of-the-art unmodified SHVC method.

5. Conclusions

In this chapter, we proposed a fast encoding method using temporal-spatial correlation and inter-layer correlation to reduce the encoding complexity for quality SHVC. In our scheme, the split quadtree information of the BL is utilized to facilitate the prediction of split CTU quadtree selection process in the ELs by avoiding redundant computations. Performance evaluations show that our approach results in significant SHVC coding complexity reduction (up to 77.74%, on average) while minimally hampering the overall bitrate.

Author details

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

*Address all correspondence to: chchwang@isu.edu.tw

Department of Electronic Engineering, I-Shou University, Kaohsiung, Taiwan

References

- [1] High Efficiency Video Coding. Rec. ITU-T H.265 and ISO/IEC 23008-2, 2013 file:///C:/Users/chchwang/Downloads/T-REC-H.265-201304-S!!PDF-E.pdf.
- [2] Advanced Video Coding for Generic Audiovisual Services. ITU-T Rec. H.264 and ISO/IEC 14496 10, ITU-T and ISO/IEC, 2010 file:///C:/Users/chchwang/Downloads/T-REC-H.264-201201-S!!PDF-E.pdf.
- [3] Joint Call for Proposals on Video Compression Technology. Kyoto, Japan, Document VCEG-AM91 of ITU-T Q6/16 and N1113 of JTC1/SC29/WG11, 2010 <http://www.itu.int/oth/T4601000002/en>.
- [4] J. Ohm, W. J. Han and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1649–1668, 2012.
- [5] B. Bross, W. J. Han, J. R. Ohm, G. J. Sullivan and T. Weingand, "High Efficiency Video Coding (HEVC) Text Specification Draft 8," JCT-VC Document, JCTVC-J1003, 2012 http://phenix.it-sudparis.eu/jct/doc_end_user/current_document.php?id=6465.
- [6] C. C. Wang, C. W. Tung and J. W. Wang, "An Effective Transform Unit Size Decision Method for High Efficiency Video Coding," Mathematical Problems in Engineering, vol. 2014, pp. 1–10, 2014.
- [7] 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, 2014 http://phenix.int-evry.fr/jct/doc_end_user/current_document.php?id=9466.
- [8] Reference Model for Mixed and Augmented Reality Defines Architecture and Terminology for MAR Applications (DOCX). MPEG. 2014-07-11. Retrieved 2014-07-26 http://mpeg.chiariglione.org/sites/default/files/files/meetings/docs/w14537_0.docx.
- [9] G. J. Sullivan et al., "Standardized Extensions of High Efficiency Video Coding (HEVC)," IEEE Journal of Selected Topics in Signal Processing, vol. 7, no. 6, pp. 1001–1016, 2013.

- [10] D. K. Kwon, M. Budagavi and M. Zhou, "Multi-Loop Scalable Video Codec Based on High Efficiency Video Coding (HEVC)," in Proceedings of the IEEE ICASSP 2013, pp. 1749–1753, 2013.
- [11] 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, 2013.
- [12] J. Boyce, Y. Ye, J. Chen and A. K. Ramasubramanian, "Overview of SHVC: Scalable Extensions of the High Efficiency Video Coding Standard," IEEE Transactions on Circuits and Systems for Video Technology, vol. 26, no. 1, pp. 20–34, 2015.
- [13] H. R. Tohidypour, M. T. Pourazad and P. Nasiopoulos, "Adaptive search range method for spatial scalable HEVC," IEEE International Conference on Consumer Electronics (ICCE), pp. 191–192, 2014.
- [14] H. R. Tohidypour, M. T. Pourazad and P. Nasiopoulos, "Content Adaptive Complexity Reduction Scheme for Quality/Fidelity Scalable HEVC," International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 1744–1748, Vancouver, Canada, May 2013.
- [15] H. R. Tohidypour, H. R. Bashashati, M. T. Pourazad and P. Nasiopoulos, "Fast mode assignment for quality scalable extension of the high efficiency video coding (HEVC) standard: a Bayesian approach", Proceedings of the 6th Balkan Conference in Informatics (BCI), pp. 61–65, Thessaloniki, Greece, Sept, 2013.
- [16] R. Bailleul, J. De Cock and R. Van De Walle, "Fast Mode Decision for SNR Scalability in SHVC Digest of Technical Papers," IEEE International Conference on Consumer Electronics (ICCE), pp. 193–194, 2014.
- [17] G. E. Qingyangl and H. U. Dong, "Fast Encoding Method Using CU Depth for Quality Scalable HEVC," IEEE Workshop on Advanced Research and Technology in Industry Applications (WARTIA), pp. 1366–1370, 2014.
- [18] L. Hahyun, K. J. Won, L. Jinho, C. J. Soo, K. Jinwoong and S. Donggyu, "Scalable Extension of HEVC for Flexible High-Quality Digital Video Content Services," ETRI Journal, vol. 35, no. 6, pp. 990–1000, 2013.
- [19] D. K. Kwon, M. Budagavi, M. Zhou, "Multi-loop scalable video codec based on high efficiency video coding (HEVC)," International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 1749–1753, Vancouver, Canada, May 2013.
- [20] David C. Lay, "Linear Algebra and Its Applications," 5th Edition, University of Maryland, College Park, Pearson Addison Wesley, 2016.
- [21] https://hevc.hhi.fraunhofer.de/svn/svn_SHVCSoftware/tags/SHM-6.0/