J Supercomput (2017) 73:1063–1084
DOI 10.1007/s11227-016-1730-y

Fast coding unit (CU) determination algorithm for high-efficiency video coding (HEVC) in smart surveillance application

Byung-Gyu Kim[1]

Published online: 5 May 2016
© Springer Science+Business Media New York 2016

Abstract High-efficiency video coding (HEVC) is a successor to the H.264/AVC standard as the newest video-coding standard using a quad-tree structure with the three block types of a coding unit (CU), a prediction unit (PU), and a transform unit (TU). This has become popular to apply to smart surveillance systems, because very high-quality image is needed to analyze and extract more precise features. On standard, the HEVC encoder uses all possible depth levels for determination of the lowest rate-distortion (RD) cost block. The HEVC encoder is more complex than the H.264/AVC standard. An efficient CU determination algorithm is proposed using spatial and temporal information in which 13 neighboring coding tree units (CTUs) are defined. Four CTUs are temporally located in the current CTU and the other nine neighboring CTUs are spatially situated in the current CTU. Based on the analysis of conditional probability values for SKIP and Merge modes, an optimal threshold value was determined for judging SKIP or Merge mode according to the CTU condition and an adaptive weighting factor. When SKIP or Merge modes were detected early, other mode searches were omitted. The proposed algorithm achieved approximately 35 % time saving with random-access configuration and 29 % time reduction with low-delay configuration while maintaining comparable rate-distortion performance, compared with HM 12.0 reference software.

Keywords High-efficiency video coding (HEVC) · Early SKIP mode detection · Inter-prediction · Optimal threshold · Conditional probability

✉ Byung-Gyu Kim
bg.kim@sm.ac.kr

[1] Department of IT Engineering, Sookmyung Women's University, Seoul, Korea

1 Introduction

The Internet of Things (IoT) was Born between 2008 and 2009, when the number of things connected to the Internet exceeded the number of people connected. By 2020, several tens of billions of devices are predicted to be connected. It is envisioned that the physical things/devices will be outfitted with different kinds of sensors and actuators and connected to the Internet via heterogeneous access networks enabled by technologies such as embedded sensing and actuating, radiofrequency identification (RFID), wireless sensor networks, real-time, and semantic Web services [1]. IoT is actually a network of networks with many unique characteristics.

Based on IoT environment, the smart surveillance technology becomes a critical component in security infrastructures and the system architecture. To accomplish scalable smart video surveillance, an inference framework in visual sensor networks is necessary, one in which autonomous scene analysis is performed via distributed and collaborative processing among camera nodes without necessity for a high-performance server [2]. Also, the main goal of smart surveillance systems is to analyze the scene focusing on the detection and recognition of suspicious activities performed by humans in the scene, so that the security personnel can pay closer attention to these preselected activities. To accomplish that, several problems have to be solved first, for instance background subtraction, person detection, tracking and re-identification, face recognition, and action recognition. Even though each of these problems have been researched in the past decades, they are hardly considered in a sequence. But each one is usually solved individually [3,4]. As rapid development of computing hardware, more high-quality networked videos such as full high definition (full HD) and ultrahigh definition (UHD) have become popular to analyze the scene and extract the contextual information more accurately. However, it is inevitable to design a fast scheme of video-encoding system to support real-time high-quality surveillance system [5].

High efficient video coding (HEVC) is the latest international video-coding standard issued by the Joint collaborative team on video coding (JCT-VC) [6], which is a partnership between the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG), two prominent international organizations that specify video-coding standards [7]. Increasing demands for high-quality full HD, UHD, and higher-resolution video necessitate bit-rate savings for broadcasting and video streaming. HEVC aims to achieve a 50 % bit-rate reduction, compared with the previous H.264/AVC video standard, while maintaining quality.

HEVC is based on a coding unit (CU), a prediction unit (PU), and a transform unit (TU). The CU is a basic coding unit analogous to the concept of macroblocks in H.264/AVC. However, a coding tree unit (CTU) is the largest CU size that can be partitioned into four sub-CUs of sizes from 64 × 64 to 8 × 8. An example of the CU partitioning structure is shown in Fig. 1. This flexible quad-tree structure leads to improved rate-distortion performance and also HEVC features for advanced content adaptability. The PU is the basic unit of inter/intra-prediction containing large blocks composed of smaller blocks of different symmetric shapes, such as square, rectangular, and asymmetric. The TU is the basic unit of transformation defined independently from the PU, but the size is limited to the CU, to which the TU belongs. Separation

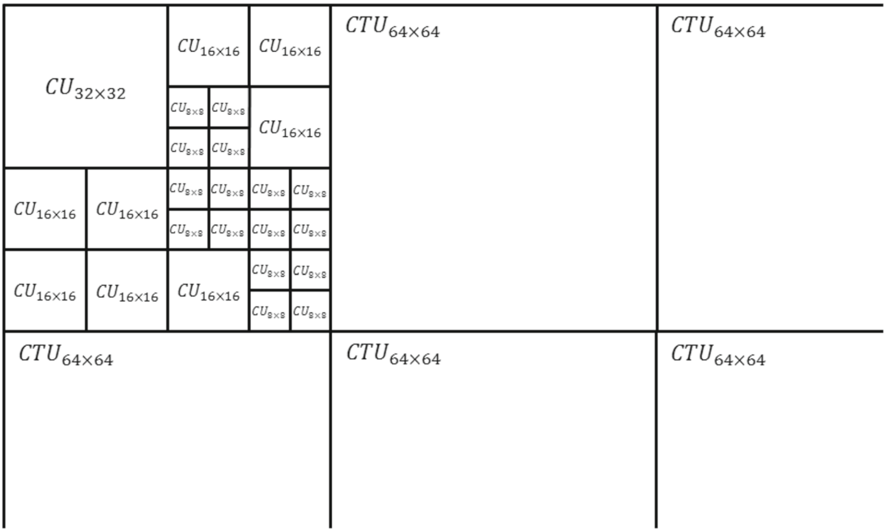

Fig. 1 An example of the CU partitioning structure

of the block structure into three different concepts allows optimization according to role, which results in improved coding efficiency [8]. However, these advanced tools cause an extremely high computational complexity. Therefore, a decrease in the computational complexity is desired.

Much effort has been exerted to reduce the complexity. Pai-Tse et al. [9] proposed a fast zero block detection algorithm based on SAD values using inter-prediction results. Features of the proposed algorithm were applied to different HEVC transform sizes. A 48 % of time saving for QP = 32 was achieved. Zhaoqing Pan et al. [10] proposed a fast CTU depth decision algorithm using the best quad-tree depth of spatial and temporal neighboring CTUs, relative to the current CTU, for an early quad-tree depth 0 decision. Correlations between the PU mode and the best CTU depth selection were also used for a depth 3 skipped decision. A 38 % of time reduction for all QPs was achieved under common testing conditions.

Hassan Kibeya et al. [11] proposed a fast CU decision algorithm for the block structure encoding process. Based on early detection of zero quantized blocks, the number of CU partitions to be searched was reduced. Therefore, a significant reduction in encoder complexity was achieved and the proposed algorithm had almost no loss of bit-rate or PSNR, compared with HM 10.0 reference software. Also there are several schemes for HEVC [12] streaming issues that may involve this new state-of-the-art codec compared to MPEG-2 streaming issues [13] and H.264/AVC streaming issues [14], as well as wireless video delivery issues [15–18].

In this study, an early SKIP and Merge mode detection algorithm is proposed using neighboring block and depth information. In Sect. 2, related researches are presented. Statistical analysis and the proposed method are presented in Sect. 3. Experimental results and discussion are shown in Sect. 4. Conclusions are presented in Sect. 5.

2 Related works

The fast encoding technique of HEVC is essential to support real-time applications. Many approaches have been reported for fast HEVC encoder design. In this section, related work is presented. In this context, there are four approaches that are generally used to achieve a fast algorithm in HEVC.

2.1 Prediction-based approach

Efficient prediction techniques are effective to achieve a fast algorithm in HEVC. A spatial and temporal correlation-based method has been proposed [19] that depends upon a correlation index of a current CTU with its spatial and temporal neighbors. The CTU is classified into the three categories of High Similarity, Medium Similarity, and Low Similarity. A CTU in the High Similarity category will have a small depth search range, a medium range in the Medium Similarity category, while a Low Similarity CTU will experience the entire depth range from 0 to 3. The algorithm provides an approximate 25 % time saving with a 0.16 % bit-rate increment.

Another method [20] adaptively determines the CU depth range using a tree block property in which small depth levels are always selected at tree blocks in homogeneous regions, and large depth levels are selected at tree blocks with active motion or rich texture, so that motion estimation can be skipped. In [21], an early CU termination algorithm has been proposed, which is commonly known as ECU. As ECU mechanism, no further processing of sub-tree is required when the current CU selects the SKIP mode as the best mode. An early skip detection (ESD) was reported by Yang et al. [22]. By checking the differential motion vector (DMV) and a coded block flag (CBF) after searching the best INTER 2N × 2N mode, it decides whether more searches are needed. After selecting the best INTER 2N × 2N mode which has the minimum RD cost, this algorithm checks the DMV and CBF. If DMV and CBF of the best INTER 2N × 2N mode are equal to (0, 0) and zero, respectively, the best mode of the current CU is determined as SKIP mode early.

Generally, bi-prediction is effective when a video contains scene changes, camera panning, zoom-in/out, and fast scenes. The RD costs of the forward and backward prediction increase when bi-prediction is the best prediction mode [23]. A bi-prediction skipping method that can efficiently reduce the computational complexity of bi-prediction is presented with the assumption that if bi-prediction is selected as the best prediction mode, the RD costs of blocks included in both the forward and backward lists can be larger than the average RD cost of previous blocks, which are coded based on forward and backward prediction.

2.2 Probabilistic approach

A modeling-based CU selection method has been proposed [24] using the statistical observation that all CUs with pyramid variance of the absolute difference (PVAD) tend to be encoded at the same depth. The CU selection process is modeled as a Markov random field (MRF) inference problem optimized using a graph cut algorithm.

Termination of CU splitting is determined based on an RD cost-based maximum a posteriori (MAP) approach. A proposed algorithm achieved a 53 % time savings with a bit-rate increment of 1.32 % [24] with a proposed extension [25] in which instead of using PVAD, only VAD was used because VAD is proportional to the RD cost similar to [24]. A binary classification modeled CU selection method has also been proposed [26]. An off-line support vector machine (SVM) model optimized using weights for training samples was used to classify a CU as either split or unsplit. Weights were generated as an RD-cost difference based on misclassification. SVM was applied for feature vectors to make the CU splitting termination decision. A 44.7 % reduction in complexity was achieved with 1.35 % bit-rate increment in the RA configuration, and a 41.9 % time savings with a 1.66 % bit-rate increment in the LD configuration.

Bayes' rule-based early CU splitting and a pruning decision method have been proposed [27]. In early split testing, Bayes' rule based on a low complexity RD cost can be used to determine whether a current CU should be split. For no splitting, a CU early pruning test is performed. A pruning decision is made based on the full RD cost of the CU. Features for both tests are periodically updated online. The method achieved an approximate 50 % encoding time reduction with only a 0.6 % increment in bit-rate.

A pyramid motion divergence (PMD)-based fast inter-CU selection method has been proposed [28] based on the statistical property that CUs with similar PMD values have the same splitting mode. Optical flow is used to find MVs so that PMD can be estimated. Optical flow estimation causes a 9 % computational burden. This method achieved an approximate 41 % time savings with an approximate 1.9 % bit-rate increment.

Based on statistical analysis, the three approaches known as SKIP mode decision (SMD), CU skip estimation (CUSE), and Early CU Termination (ECUT) have been proposed [29]. SMD is used to determine if the remaining modes, except for the SKIP mode, are performed or not. CUSE and ECUT are used to determine if large and small CU sizes are coded, respectively. Thresholds for SMD, CUSE, and ECUT are designed based on Bayes' rule with a complexity factor. An update process is performed for estimation of statistical parameters for SMD, CUSE, and ECUT considering the characteristic RD cost. Experimental results demonstrated that the proposed CU size decision algorithm significantly reduced the computational complexity by 69 %, on average, with a 2.99 % BDBR increase.

2.3 Advanced RD-cost calculation approach

An advanced RD-cost calculation method concentrating on rate distortion optimization (RDO) was proposed [30]. Top SKIP and Early Termination were used to reduce the number of RD-cost estimations. For Top SKIP, the starting depth and larger depths of 64 × 64 and 32 × 32 are skipped if high correlation exists between the minimum depth of the current CTB and the collocated CTB in the previous frame. Early Termination is a process that is complementary to Top SKIP in which the CU splitting process is terminated if the best RD cost for the current CU is lower than a threshold value that is adaptively calculated using both spatial and temporal features. This method reduced the encoding time by up to 45 % with a negligible quality loss within 0.1dB.

Apart from fast mode decision algorithms, research has focused on improving the rate-distortion calculation technique. In this context, a mixed Laplacian-based RD-cost calculation scheme has been proposed [31] in which inter-predicted residuals exhibited different statistical characteristics for CU blocks in different depth levels. Based on the mixed Laplacian distribution, experimental results showed that the proposed rate and distortion models were capable of better estimation of actual rates and distortions than a single Laplacian distribution.

3 Proposed algorithm

3.1 Problem and motivation

Some fast algorithms can be divided into the three parts of CU splitting termination, fast prediction unit (PU) methods, and fast transform schemes. CU splitting termination has a quad-tree structure and recursively performs the 64 × 64 size to the 8 × 8 size. If the best CU can be determined, only essential CUs are split [32] with the best mode information in the upper level. This approach has a low bit ratio, because detection conditions are intuitive. However, if the condition boundary is widened, the quality loss is increased greatly. Also, it is difficult to select the best mode because the characteristic of the best mode is easily changeable. Therefore, unique characteristics are required for a particular mode. An encoder performs a mode decision process using the most common prediction mode after a size decision for the CU in the process of CU splitting. If the best mode can be predicted in advance, the search time for the mode decision process can be reduced. ECU is used for this approach. However, these types of algorithms are incompatible with other algorithms due to omission of the existing routine in advance.

The TU process recursively partitions the structure of a given prediction block into transform blocks that are represented by the residual quad-tree (RQT). An RD-cost evaluation is performed multiple times within each quad-tree structure as 1 time for a 32 × 32 TU, 4 times for a 16 × 16 TU, and 16 times for an 8 × 8 TU. Early Termination allows only performing the TU size that has been selected as the best size to effectively reduce time consumption. However, this approach requires many RD-cost computations. Therefore, it is suitable for use in a supporting role rather than performing a single algorithm. These problems due to use of a quad-tree structure as the most common prediction mode and transformation in RQT can be solved by avoiding unnecessary processes and the number of executions can be reduced significantly. However, there is room for improvement using spatial and temporal information for detection of SKIP and Merge modes to achieve speedup of the encoding system.

Most real-world video sequences contain a substantial amount of background and many motionless objects with high temporal correlations between successive frames and spatial correlations among adjacent pixels. Generally, these kinds of motionless regions are encoded as SKIP or Merge mode. If a CB is encoded as SKIP or Merge

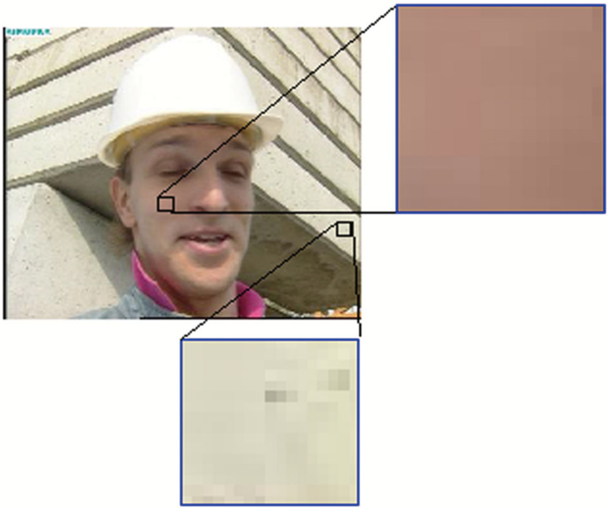

Fig. 2 An example of spatial redundancy

mode, then only the SKIP flag and corresponding Merge index are transmitted to the decoder side, which ensures that SKIP and Merge mode requires a minimum number of bits. Typical video contents include large homogenous regions and many motionless objects. These characteristics are described in Fig. 2.

When a CB is encoded as SKIP mode in HEVC, two conditions should be satisfied. One is that the motion vector difference between the current 2N×2N PU and the neighboring PU is zero, and the another is that residuals are all quantized to zero. The CTU structure and mode decision result of a video frame from the Cactus sequence is shown in Fig. 3 where green areas represent intra-mode and red represents inter-mode. Blue and gray regions describe PU and TU splits, respectively. Yellow areas indicate blocks that are encoded as SKIP mode. Most of the stationary regions of the video frame are encoded as SKIP mode Fig. 2.

Large portions of regions that are encoded as SKIP mode exist. To distinguish these regions as SKIP or Merge mode, an efficient algorithm is needed prior to the beginning of the RD-cost calculation process. Further statistical analysis was performed using different sequences and QP values to determine CUs as SKIP. In depth level 0, the portion of SKIP as the best mode was 81.37 % on average. With a depth level of 1, 66.46 % SKIP mode was determined, on average (in Table 1).

When the depth level was deeper, the amount of SKIP mode decreased. When QP value was high, the portion of SKIP mode increased. Although the BQTerrace and BasketballPass sequences have more complicated motion activities than other sequences, they provide a sufficient amount of SKIP mode. Also, the characteristic that stationary blocks tend to be positioned around each other was observed. The development of the proposed algorithm was motivated by this observation and analysis.

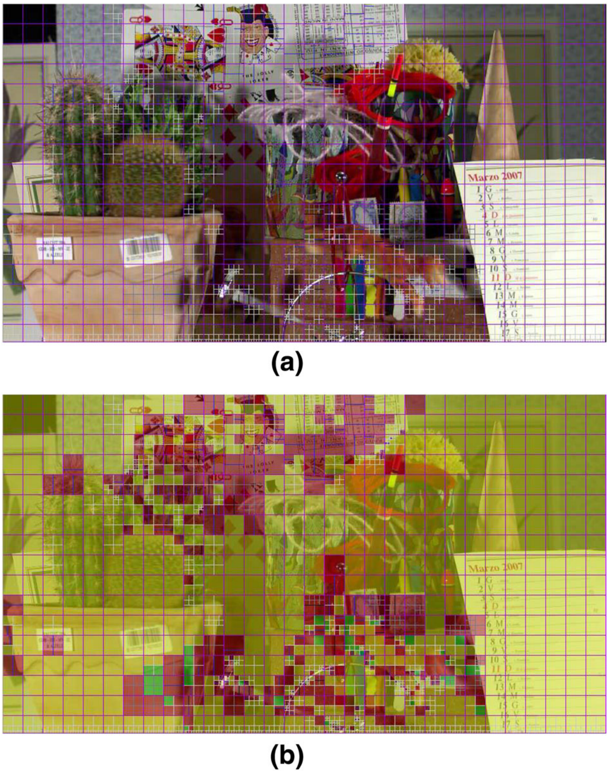

Fig. 3 The CTB structure and the corresponding CUs which are finally encoded as SKIP mode in the Cactus for QP 32. **a** The CTB structure of frame no. 35 and **b** the CUs which are encoded as SKIP mode of frame no. 35 are shown using *yellow color* (color figure online)

3.2 An early SKIP and Merge mode detection method

In natural pictures, neighboring blocks usually exhibit similar textures. Optimal inter-prediction of a current block may be strongly correlated with adjacent blocks. Homogenous blocks, which are encoded as SKIP or Merge mode, especially tend to be located adjacent to each other (Table 1).

Based on this consideration, the conditional probability $P(O|NC)$ was analyzed, where neighboring CTUs (NC) represented the event that a current CTU was encoded as depth 0 and SKIP mode, CBF, and Merge mode flag were true with NC = A, B, C, D, E, F, G, H, I, J, K, L, M corresponding to left, above-left, above, above-right, collocated-current, collocated-left, collocated-above-left, collocated-above, collocated-above-right, collocated-right, collocated below-right, collocated-below, and collocated-below-left as shown in Fig. 4. Also, O indicated the event that the current CTU was encoded as depth 0 and flags were true. Thirteen defined blocks as CTUs from A to M were used in Fig. 4. B and C class sequences were used based on the degree of motion. The conditional probabilities of adjacent CTUs were skipped when the current CTU was skipped (Tables 2, 3, 4).

Table 1 Percentage of CUs that are encoded as SKIP for different video sequences with different QP values

Sequence	QP	% SKIP mode for different CU size			
		64 × 64	32 × 32	16 × 16	8 × 8
Traffic (2560 × 1600)	22	79	60	46	35
	27	85	64	51	36
	32	89	68	52	33
	37	93	70	48	28
	Avg.	86.50	65.50	49.25	33.00
ParkScene (1920 × 1080)	22	70	64	45	30
	27	82	67	51	35
	32	88	69	54	37
	37	92	69	51	40
	Avg.	83.00	67.25	50.25	35.50
BQTerrace (1920 × 1080)	22	68	47	64	26
	27	82	68	51	39
	32	91	71	60	46
	37	94	74	66	50
	Avg.	83.75	65.00	60.25	40.25
PartyScene (832 × 480)	22	87	57	33	20
	27	86	64	43	25
	32	86	66	46	27
	37	90	64	46	28
	Avg.	87.25	62.75	42.00	25.00
BlowingBubbles (416 × 240)	22	51	46	36	23
	27	56	53	42	28
	32	72	59	48	30
	37	75	62	54	28
	Avg.	63.50	55.00	45.00	27.25
BasketballPass (416 × 240)	22	82	82	61	33
	27	84	83	66	36
	32	84	83	69	35
	37	87	85	72	32
	Avg.	84.25	83.25	67.00	34.00
Total average		81.37	66.46	52.29	32.50

The temporal CTUs A, B, C, and D had probability values from 76.0 to 77.1 %, and the spatial CTUs E, F, G, H, I, J, K, L, and M had probability values from 69.6 to 73.2 %, on average, in the class B Cactus sequence (Table 2). Also, in class B BQTerrace sequence, temporal CTUs had probability values from 78.5 to 80.1 % and spatial CTUs had probability values from 58.6 to 62.8 %. Also, each of these probabilities tended to follow the above trends from Tables 2, 3, and 4.

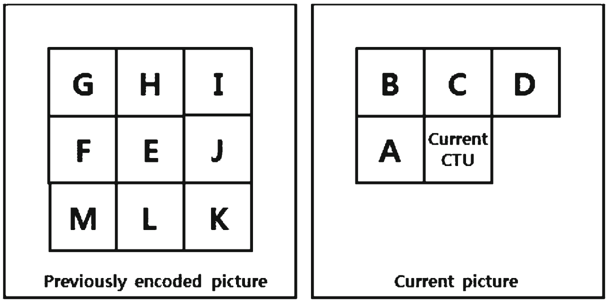

Fig. 4 Neighboring CTUs and its collocated CTUs in a previously encoded frame

From experimental results, when reference CTUs were encoded as SKIP mode more than as Merge mode in depth level 0, the current CTU had a high probability to be encoded as SKIP or Merge mode, especially when the spatial CTUs A, B, C, and D, which are spatially located next to current CTU, were used. Also, the temporal CTUs E, F, G, H, I, J, K, L, and M were temporally located next to the current CTU. Based on these observations, the two groups of spatial CTUs and temporal CTUs were defined. Then, weighting factors were determined for each group and an equation was defined as:

$$\mathrm{ETC} = \sum_{n \in NC} C_n \lambda_n \geq \alpha, \qquad (1)$$

where ETC indicates Early Termination. When the equation is satisfied, other inter-prediction processes are omitted. C_n represents the availability of neighboring CTUs. When CTU has SKIP flag as true or CBF is 0 or Merge flag is true, C_n is set to 1. Otherwise, C_n is 0. λ_n is a weighing factor based on a conditional probability. For spatial CTUs, λ_n is set to 1.0 and for temporal CTUs it is set to 0.75. The equation is only one form, but the three thresholds of T_{SKIP}, T_{CBF}, and T_{Merge} can be obtained with individual flags: T_{SKIP} using the SKIP flag, T_{CBF} using the CBF flag, and T_{Merge} using the Merge flag. Therefore, each has a separate threshold value. To obtain the optimal value of α, additional analysis was performed (Table 5). The optimal threshold value of α was determined to be 4.

3.3 Overall procedure

The overall procedure and flow of the proposed algorithm are shown in Fig. 5. First, the inter-prediction stage starts after the depth of the current block is checked. When the depth is 0, we go to the next step; otherwise, original regular routines are processed. Next, 3 Early Termination Conditions (ETCs) are calculated. $\mathrm{ETC}_{\mathrm{SKIP}}$ is intended for SKIP mode based on the SKIP flag. $\mathrm{ETC}_{\mathrm{CBF}}$ is also used for SKIP; however, the coded block flag (CBF) is used to check the KIP mode. Lastly, $\mathrm{ETC}_{\mathrm{Merge}}$ is intended for Merge mode based on the Merge flag. ETCs are determined based on Eq. 1. Then, $\mathrm{ETC}_{\mathrm{SKIP}}$ is compared with α.

Table 2 Results of the conditional probability P(O|NC) with SKIP flag

Class	Sequence	QP	Statistical results of the conditional probability P(O|NC) with SKIP												
			A	B	C	D	E	F	G	H	I	J	K	L	M
B	Cactus	22	67.7	67.6	70.3	66.5	67.4	66.2	61.3	66.6	66.1	67.2	63.4	64.3	62.1
		27	75.5	74.6	77.8	74.0	73.3	70.4	69.5	72.0	71.0	71.4	70.7	71.8	68.2
		32	79.2	78.4	80.6	78.4	74.6	73.2	72.2	72.6	72.3	73.5	73.4	72.8	71.7
		37	85.9	85.5	86.9	84.9	77.6	76.9	75.9	76.2	76.2	77.2	77.8	77.4	76.4
		Avg.	77.1	76.5	78.9	76.0	73.2	71.7	69.7	71.8	71.4	72.3	71.3	71.6	69.6
	BQTerrace	22	64.1	62.6	64.6	60.7	41.3	43.3	47.7	46.7	44.1	39.8	35.3	37.9	40.5
		27	74.9	73.3	74.8	73.8	54.4	53.5	53.2	54.9	53.1	53.3	49.9	50.3	50.4
		32	88.1	86.6	87.2	86.8	71.1	71.3	69.6	69.8	69.0	70.1	68.8	70.8	71.3
		37	93.4	92.6	93.0	92.6	81.1	81.2	80.0	79.8	78.7	80.0	80.3	81.8	81.7
		Avg.	80.1	78.8	79.9	78.5	62.0	62.3	62.6	62.8	61.2	60.8	58.6	60.2	61.0
C	BQMall	22	92.1	87.0	90.5	87.5	88.2	93.3	85.4	85.4	84.3	86.4	90.9	86.3	88.2
		27	94.7	93.3	95.0	94.2	88.8	87.4	92.0	91.3	90.5	87.1	83.3	84.6	81.8
		32	94.4	96.2	95.0	94.4	88.9	95.7	88.6	88.7	88.8	88.0	86.0	86.5	84.2
		37	97.8	98.3	97.7	97.1	96.2	91.2	97.1	96.6	96.0	94.0	93.9	95.1	94.3
		Avg.	94.7	93.7	94.6	93.3	90.5	75.0	90.7	90.5	89.9	88.9	88.5	88.1	84.1
	PartyScene	22	94.8	90.0	99.2	85.1	60.3	59.6	57.6	58.3	55.7	59.3	67.4	66.7	65.9
		27	89.7	90.5	90.8	86.9	87.7	85.3	81.5	80.8	77.0	79.8	77.9	86.7	83.8
		32	91.2	91.5	90.3	37.5	69.8	71.9	69.7	71.0	64.4	65.2	68.3	67.1	66.7
		37	93.1	91.3	93.7	85.7	67.5	67.6	68.9	68.8	63.5	66.7	72.4	65.5	64.0
		Avg.	92.2	90.8	93.5	73.8	71.4	71.1	69.4	69.7	65.2	67.8	71.5	71.5	70.1

Table 3 Results of the conditional probability P(O|NC) with CBF

Class	Sequence	QP	Statistical results of the conditional probability P(O|NC) with CBF												
			A	B	C	D	E	F	G	H	I	J	K	L	M
B	Cactus	22	87.9	88.3	90.0	87.6	87.7	84.7	82.9	86.9	86.0	86.4	86.2	86.1	83.1
		27	88.4	88.3	90.1	87.8	82.1	80.8	80.0	81.3	81.0	81.1	80.0	80.6	79.1
		32	91.6	91.3	91.9	91.4	83.8	83.3	83.0	82.8	82.7	83.5	83.6	83.1	82.5
		37	95.4	95.3	95.8	95.2	86.1	86.9	86.4	85.3	85.1	86.1	86.3	86.3	87.0
		Avg.	90.8	90.8	91.9	90.5	84.9	83.9	83.1	84.1	83.7	84.3	84.0	84.0	82.9
	BQTerrace	22	86.6	84.5	86.7	84.1	60.8	59.3	62.3	62.4	64.4	60.2	55.7	56.1	56.9
		27	93.6	92.8	93.7	93.1	69.5	68.9	70.0	71.7	71.0	68.3	66.0	67.1	67.8
		32	98.1	97.5	97.6	97.7	85.4	85.5	84.4	84.7	84.3	84.6	85.4	86.2	86.4
		37	98.7	98.3	98.5	98.1	91.3	91.3	90.4	90.7	90.0	90.5	90.7	91.5	91.8
		Avg.	94.3	93.3	94.1	93.2	76.8	76.2	76.8	77.4	77.4	75.9	74.4	75.2	75.7
C	BQMall	22	90.7	91.0	93.1	92.2	94.1	93.3	91.6	95.8	94.1	91.8	90.9	95.4	90.1
		27	91.5	88.9	91.8	88.8	94.4	93.7	88.6	93.4	86.3	88.4	87.0	92.3	93.1
		32	90.1	92.5	94.3	93.1	93.5	90.5	89.4	90.2	90.2	95.4	94.9	89.0	95.3
		37	92.3	91.1	93.6	93.6	94.7	91.5	92.7	92.6	92.7	94.7	96.9	97.0	97.7
		Avg.	91.1	91.0	93.2	91.9	94.2	92.2	90.6	93.0	90.8	92.6	92.4	93.4	93.9
	PartyScene	22	95.4	92.5	94.4	89.7	65.5	64.9	62.7	63.3	60.7	64.4	69.6	71.1	70.5
		27	98.2	98.9	99.7	98.1	99.1	98.0	99.1	99.2	97.8	97.7	97.9	100.0	98.6
		32	97.2	97.3	98.0	93.4	90.6	91.7	91.9	91.6	89.8	91.3	91.5	90.8	91.3
		37	96.6	97.8	98.0	93.0	77.9	77.5	78.4	78.8	78.8	79.0	77.6	78.2	76.0
		Avg.	96.8	96.6	97.5	93.6	83.3	83.0	83.0	83.2	81.8	83.1	84.1	85.0	84.1

Table 4 Results of the conditional probability P(O|NC) with Merge flag

Class	Sequence	QP	Statistical results of the conditional probability P(O|NC) with Merge flag												
			A	B	C	D	E	F	G	H	I	J	K	L	M
B	Cactus	22	86.9	86.9	87.6	87.5	83.1	85.4	82.2	84.2	84.6	84.7	83.4	84.7	83.1
		27	92.0	92.3	92.7	91.5	88.2	87.8	87.0	87.0	86.8	87.8	87.4	88.2	87.1
		32	92.3	92.5	92.4	91.8	89.1	88.5	87.9	87.5	86.7	87.8	88.7	88.8	88.6
		37	93.7	93.5	94.1	93.8	87.3	87.4	86.4	86.0	85.8	87.5	88.6	88.1	87.6
		Avg.	91.2	91.3	91.7	91.2	86.9	87.3	85.9	86.2	86.0	87.0	87.0	87.4	86.6
	BQTerrace	22	84.6	83.8	83.8	82.0	51.6	53.1	58.2	57.4	58.6	51.9	51.2	51.4	53.0
		27	89.1	87.7	88.0	87.7	76.0	75.1	73.5	75.6	72.1	69.5	68.7	72.5	73.9
		32	92.1	92.9	93.0	92.8	79.5	79.5	78.5	78.9	77.9	78.1	79.3	80.1	80.6
		37	95.4	94.9	95.0	94.8	83.8	84.0	82.8	82.7	81.6	82.8	83.9	84.7	84.7
		Avg.	90.3	89.8	90.0	89.3	72.7	72.9	73.3	73.6	72.6	70.6	70.8	72.2	73.0
C	BasketballDrill	22	83.1	79.5	82.2	80.0	78.9	75.0	69.6	73.0	73.9	75.0	67.1	72.1	78.0
		27	82.7	82.0	85.4	79.4	83.0	76.2	73.5	78.4	76.1	76.9	76.4	75.3	67.1
		32	90.4	82.7	43.8	86.8	89.6	85.4	77.4	81.9	79.7	83.1	84.8	84.8	81.4
		37	86.9	85.7	89.6	82.9	84.9	82.4	76.2	82.0	77.6	80.3	80.2	81.8	80.4
		Avg.	85.8	82.5	75.3	82.3	84.1	79.7	74.2	78.8	76.8	78.8	77.1	78.5	76.7
	PartyScene	22	99.3	98.8	98.3	92.8	81.0	80.7	81.4	81.7	78.7	81.4	80.4	80.0	79.5
		27	93.4	94.2	93.5	91.4	91.2	90.2	86.1	85.8	83.7	86.9	87.4	91.6	91.9
		32	94.4	96.0	93.5	89.3	81.1	84.4	82.8	81.3	78.0	78.3	88.9	82.9	85.5
		37	97.7	97.3	98.0	93.5	84.4	83.1	82.4	85.0	83.5	84.0	79.3	78.2	76.0
		Avg.	96.2	96.5	95.8	91.8	84.5	84.6	83.2	83.5	81.0	82.6	84.0	83.2	83.2

Table 5 Average encoding results for different values of α

Class	Evaluation items	α								
		2	2.75	3	3.5	3.75	4	4.5	4.75	5.5
B	BDBR (%)	5.6	2.8	3.2	2.5	2.8	1.1	1.9	1.0	1.1
	Bit-rate (dB)	0.8	0.1	0.08	0.0	−0.1	−0.3	−0.1	−0.4	−0.3
	PSNR (dB)	−0.3	−0.1	−0.2	−0.1	−0.1	−0.1	−0.1	−0.1	−0.1
	TS (%)	−47.8	−42.4	−43.2	−41.4	−39.3	−44.7	−39.4	−43.3	−39.4
C	BDBR (%)	7.3	4.5	3.8	3.9	3.1	1.3	2.8	1.4	1.27
	Bit-rate (dB)	3.1	1.5	1.3	1.3	0.8	0.1	0.7	0.0	0.07
	PSNR (dB)	−0.5	−0.4	−0.3	−0.3	−0.2	−0.1	−0.2	−0.1	−0.1
	TS (%)	−35.1	−32.5	−31.1	−30.7	−27.6	−28.3	−30.4	−28.8	−28.1

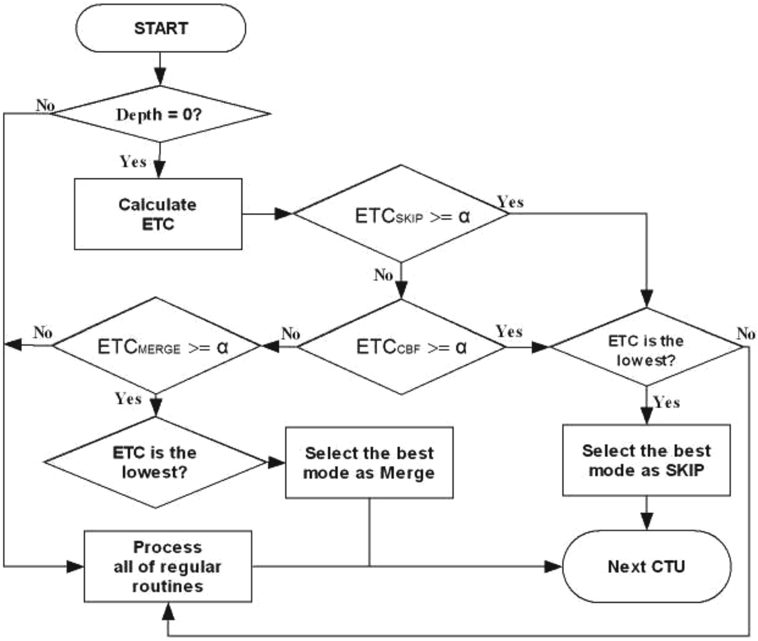

Fig. 5 Overall procedure of the proposed algorithm

Experimentally, the optimal α value was determined to be 4. When ETC_{SKIP} is greater than or equal to α, we move to the final stage. If not, original regular routines are processed. The final stage is determination of whether the ETC is the lowest value among neighboring CTUs that have SKIP flag as true. When the ETC is lowest, the best mode for the current CTU is determined as SKIP mode, and then we move to encode the next CTU. However, if not, regular routines are processed. ETC_{CBF} and $\text{ETC}_{\text{Merge}}$ use the same procedure except for the use of individual flags. Calculation of ETCs requires $\mathrm{ETC}_{\mathrm{SKIP}}$ to use the SKIP flag, $\mathrm{ETC}_{\mathrm{CBF}}$ to use CBF, and $\mathrm{ETC}_{\mathrm{Merge}}$ to use the Merge flag for detection of the Merge mode.

4 Simulation result and evaluation

4.1 Test conditions

The proposed algorithm was tested and evaluated on HM reference software, version 12.0. All experiments were conducted using an Intel Core (TM) i7-3770 @ 3.4 GHz with 16GB of RAM. The test conditions were

- For each test sequence, 50 frames were encoded.
- The group of picture (GOP) size was set to 8.
- A value of 64 was used for the search range.
- All experiments were performed for the four quantization parameters (QP) of 22, 27, 32, and 37.
- Both random access (RA-Main) and low delay (LB-Main) profiles were considered.
- Previously defined configurations [34] were used.
- The fast encoder setting (FEN), early CU setting (ECU), and CBF fast mode setting (CFM), adaptive motion search range (ASR), and Early SKIP detection setting (ESD) were OFF.
- Test sequences which different resolutions and belonging to different classes were used (Table 6).
- Class A was not used for the LB-Main profile [34].

4.2 Performance evaluation

The proposed algorithm was evaluated using HM reference software version 12.0. The four parameters were (1) time required to encode a video sequence, (2) the number of bits in the encoded bit-stream, (3) the corresponding peak signal-to-noise-ratio (PSNR) value, and (4) BDBR. Bit-rate, PSNR, TS, and BDBR values were calculated for comparison of the proposed algorithm with HM 12 as:

$$\Delta\text{Bit-rate} = \frac{\text{Bit-rate}_{\text{proposed}} - \text{Bit-rate}_{\text{original}}}{\text{Bit-rate}_{\text{original}}} \times 100\,\%, \tag{2}$$

$$\Delta\mathrm{PSNR}_Y = \mathrm{PSNR}_Y^{\text{proposed}} - \mathrm{PSNR}_Y^{\text{original}}, \tag{3}$$

$$\Delta T = \frac{\text{Time}_{\text{proposed}} - \text{Time}_{\text{original}}}{\text{Time}_{\text{original}}} \times 100\,\%, \tag{4}$$

where ΔBit-rate indicates the total bit-rate change (percentage), $\Delta\mathrm{PSNR}_Y$ indicates the PSNR_Y changes, and ΔT is the time-saving factor (percentage). For bit-rate, a positive value indicated an increase in the number of bits and a negative value indicated a decrease. For $\Delta\mathrm{PSNR}_Y$, an improvement was a positive value and a negative value

Table 6 Test sequence description

Class	Resolution	Sequence	Frame rate (fps)	Descrption
A	2560 × 1600	Traffic	30	Medium motion with rich texture
		PeopleOnStreet	30	Medium motion with rich texture
B	1920 × 1080	BasketballDrive	50	High motion with rich texture
		Cactus	50	Medium motion with rich texture
		Kimono	24	Medium motion with rich texture
		BQTerrace	60	Medium motion with rich texture and camera movement
C	832 × 480	BasketballDrill	50	High motion
		PartyScene	50	Medium motion with zoom-in effect
		BQMall	60	Medium motion with rich texture and camera movement
		RaceHorses	30	Medium motion with rich texture
D	416 × 240	BlowingBubbles	50	Medium motion with zoom out
		BQSquare	60	Synthetic with camera movement
		BasketballPass	50	High motion with rich texture
		RaceHorses	30	Medium motion with rich texture

indicated degradation. A negative value for the time-saving factor indicated that the algorithm consumed less time than the original reference software.

The experimental results are shown in Tables 7 and 8 for RA-main and LB-main profiles, respectively. The proposed algorithm provided, on average, 35.41 and 36.88% encoding time reductions (ΔT) for RA-Main and the LB-Main profiles, respectively, with a similar perceptual video quality. Bjontegaard Delta (BD) rate values are shown in the BDBR value, including both BD-PSNR and BD-Bit-rate [33] and all tests were carried out under common test conditions and configurations [34].

For the RA-Main configuration, the Traffic, ParkScene, Cactus, BasketballDrive, BQTerrace, BasketballPass, and BQSquare sequences exhibited good time savings with a trivial quality loss. Even with a stationary background with a high degree of motion in all of these sequences, the proposed algorithm achieved good performance. The BQMall Sequence with a rich texture, camera movement, and many objects exhibited good time savings, but with a higher degree of quality loss. The PeopleOnStree and RaceHorses sequences exhibited time saving with a trivial quality loss. The PeopleOnStreet sequence contains many objects with irregular movement. The background in the RaceHorses sequence is erratic and the sequence contains camera movement. Little SKIP mode was used in this sequence.

For the LD-Main configuration, the results were similar, but quality loss was larger. Thus, the proposed scheme is appropriate for use with the RA-Main configuration. The proposed algorithm achieved an 18.06–49.52 % of time reduction with a 0.6–1.3 % of BDBR for the RA-Main configuration with class A sequences. For class B sequences, a 28.92–45.38 % of time reduction with a 0.3–1.2 % of BDBR increment under the RA-Main profile was achieved, and a 22.48–42.04 % time savings with

Table 7 Performance of the proposed algorithm compared with HM 12.0 with RA-MAIN configuration

Class	Sequences	ΔBit-rate (%)	ΔPSNR_Y (dB)	ΔT (%)	BDBR (%)
A	Traffic	−0.75	−0.06	−49.52	1.3
	PeopleOnStreet	−0.28	−0.05	−18.06	0.6
B	Kimono	−0.24	0.03	−28.92	0.4
	ParkScene	−0.47	−0.05	−45.38	1.0
	Cactus	−0.41	−0.03	−38.21	0.8
	BasketballDrive	−0.27	−0.02	−34.26	0.3
	BQTerrace	−0.88	−0.05	−45.50	1.2
C	BasketballDrill	−0.36	−0.03	−27.00	0.3
	BQMall	−0.07	−0.09	−30.83	2.0
	PartyScene	−0.26	−0.07	−24.27	0.8
D	BasketballPass	−0.41	−0.06	−40.92	0.5
	BQSquare	−0.73	−0.05	−43.57	0.4
	BlowingBubbles	−0.37	−0.04	−24.07	0.6
	RaceHorses	−0.25	−0.05	−16.10	0.6
Average		−0.39	−0.04	−35.41	0.82

Table 8 Performance of the proposed algorithm compared with HM 12.0 for the LD-MAIN configuration

Class	Sequences	ΔBit-rate (%)	ΔPSNR_Y (dB)	ΔT (%)	BDBR (%)
B	Kimono	0.03	−0.02	−22.48	0.6
	ParkScene	0.09	−0.09	−42.04	3.5
	Cactus	−0.41	−0.03	−35.80	1.3
	BasketballDrive	0.08	−0.03	−27.26	1.4
	BQTerrace	−0.58	−0.10	−42.50	3.4
C	BasketballDrill	0.43	−0.03	−22.74	1.2
	BQMall	1.25	−0.09	−26.34	4.6
	PartyScene	0.11	−0.04	−15.64	0.7
D	BasketballPass	−0.33	−0.03	−36.83	0.1
	BQSquare	−0.69	−0.05	−27.36	0.2
	BlowingBubbles	−0.07	−0.03	−14.87	0.5
	RaceHorses	−0.03	−0.02	−12.01	0.2
Average		−0.01	−0.04	−29.99	2.2

a 0.6–3.4 % of BDBR increment under LD-Main configuration was achieved. For class C sequences, a 26.86–30.90 % of encoding time saving with a 0.1–0.5 % BDBR increment under the RA-Main configuration was achieved, and a 15.64–26.34 % of time reduction with 0.7–4.6 % of BDBR increment under the LD-Main configuration was achieved. In class D sequences, a 16.10–43.57 % of time reduction with a 0.4–0.6 % of BDBR increment under the RA-Main configuration was achieved. In the

Table 9 Performance of various algorithms compared with HM 12.0 for the RA-MAIN profile

Class	Sequences	ECU [21]		ESD [22]		Sampaio's [35]		Yang's [36]		Proposed	
		ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)
A	Traffic	−48.4	0.5	−36.76	0.2	−23.26	0.8	−21.59	0.1	−49.20	1.3
	PeopleOnStreet	−17.5	0.4	−16.84	0.2	−24.14	1.2	−9.45	0.2	−18.06	0.6
B	Komono	−28.78	0.3	−23.59	0.2	−18.80	0.4	−16.35	−0.1	−28.92	0.4
	ParkScene	−44.81	0.6	−34.65	0.2	−20.51	0.6	−23.35	0.2	−45.38	1.0
	Cactus	−37.57	0.5	−29.55	0.2	−22.54	0.8	−18.70	0.2	−38.21	0.8
	BasketballDrive	−33−91	0.1	−27.24	0.2	−20.20	0.8	−19.12	0.1	−34.26	0.3
	BQTerrace	−44.66	0.8	−34.09	0.3	−20.67	0.7	−23.50	0.2	−45.50	1.2
C	BasketballDrill	−29.42	0.1	−20.76	0.1	−21.91	0.5	13.12	0.1	−27.00	0.3
	BQMall	−30.90	0.5	−17.84	0.2	−23.49	1.5	−14.38	0.1	−30.83	2.0
	PartyScene	−26.86	0.4	−12.91	0.2	−21.52	0.5	−11.58	0.1	−24.27	0.8
D	BasketballPass	−39.26	0.5	−29.33	0.2	−23.36	1.1	−19.96	−0.1	−40.92	0.5
	BQSquare	−41.09	0.3	−32.94	0.0	−24.88	0.3	−23.27	−0.2	−43.57	0.4
	BlowingBubbles	−20.34	0.6	−36.57	0.4	−25.59	0.8	−15.49	0.2	−24.07	0.6
	RaceHorses	−10.88	0.6	−19.62	0.5	−28.67	1.9	−11.74	0.4	−16.10	0.6
Average		−28.90	0.06	−21.31	0.89	−23.69	0.7	−23	0.5	−35.41	0.82

Table 10 Performance of various algorithms compared with HM 12.0 for the LD-MAIN profile

Class	Sequences	ECU [21]		ESD [22]		Sampaio's [35]		Yang's [36]		Proposed	
		ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)
B	Komono	−19.34	0.1	−17.86	0.2	−21.77	0.1	−13.61	0.1	−22.48	0.6
	ParkScene	−33.29	0.4	−28.45	0.3	−23.93	0.4	−20.09	0.2	−42.04	3.5
	Cactus	−29.23	0.3	−23.53	0.4	−22.95	0.4	−16.05	0.2	−35.80	1.3
	BasketballDrive	−22.60	0.3	−19.46	0.4	−20.13	0.5	−12.23	0.2	−27.26	1.4
	BQTerrace	−37.61	0.2	−29.71	0.3	−21.65	0.5	−21.51	0.2	−42.50	3.4
C	BasketballDrill	−21.57	−0.2	−15.35	−0.1	−20.55	0.4	−11.75	−0.1	−22.74	1.2
	BQMall	−25.71	0.2	−21.54	0.3	−23.60	0.8	−13.60	0.1	−26.34	4.6
	PartyScene	−14.70	0.1	−14.79	0.3	−22.60	0.2	−8.71	0.2	−15.64	0.7
D	BasketballPass	−10.23	0.1	−27.11	0.3	−25.23	0.7	−17.06	0.3	−36.83	0.1
	BQSquare	−35.07	0.1	−24.54	0.6	−24.65	0.3	−17.36	0.2	−27.36	0.2
	BlowingBubbles	−23.57	0.1	−14.59	0.7	−25.96	0.8	−13.37	0.5	−14.87	0.5
	RaceHorses	−13.15	0.5	−10.35	0.1	−29.61	0.9	−11.47	0.2	−12.01	0.2
Average		−23.83	0.05	−25.03	0.3	−23.79	0.57	−17.93	0.13	−29.99	2.2

LD-Main configuration, 12.01–36.83 % of time-saving factor with a 0.1–0.5 % of BDBR increment was achieved.

Tables 9 and 10 show the performance comparison with the existing methods, respectively. The proposed algorithm achieved more time reduction than other well-known algorithms with considerable loss. For ECU, unnecessary recursive routines were used. However, the proposed algorithm detected many SKIP and Merge modes at a depth level of 0. Thus, the proposed algorithm reduced the number of unnecessary recursive routines. It means that the proposed algorithm can reduce unnecessary routines. For ESD [22], the reported method processed all prediction stages in the beginning step. However, the proposed algorithm decided a proper CU node before the prediction stage started. Thus, the algorithm resulted in more speedup than for ESD [22].

When compared with Sampaio's method [35], 12 % of average time-saving factor was achieved with similar BDBR under the RA-Main configuration. For Yang's method [36], a similar performance was observed (about 12 % of time-saving factor compared to Yang's method). ECU [21] seems good in terms of BDBR performance.

In case of the LD-Main configuration, the proposed algorithm achieved about 30 % of average time-saving factor while keeping 2.2 % of BDBR performance. For BQMall sequence (Class C), the suggested algorithm gave a little large BDBR loss. In other sequences, we can see that the proposed scheme achieves a reliable performance. In case of Sampaio's and Yang's algorithms, 23.79 and 17.93 % of time reductions were observed in the average encoding time, respectively. Compared to Sampaio's method [35], the proposed algorithm yielded over 16 % of time reduction. For Yang's algorithm [36], up to 24 % of time-saving factor was achieved. From the results, we are able to deduce that the proposed algorithm is very effective in the HEVC encoding system.

5 Conclusions

An efficient CU determination algorithm was proposed using neighboring block information for the smart surveillance system. The proposed algorithm was based on the characteristics of natural video sequences using conditional probability values between a current block and adjacent blocks. The SKIP flag and CBF were used to detect the SKIP mode. The Merge flag was examined to identify the Merge mode. When adjacent CTUs were selected as SKIP or Merge mode, further processes were omitted.

One of the most important challenges for fast algorithm development in HEVC is a trade-off between video quality and encoding time reduction. An encoding time reduction of more than 35 % in random-access configuration and a 29 % encoding time saving in low-delay configuration with a negligible video quality loss and bit-rate increment were achieved. The proposed scheme can support to make real-time HEVC video system in the networked smart surveillance system. For further work, we should consider a fast prediction unit (PU) decision scheme to obtain more spedup gain, including a selection of transform unit (TU) size, adaptively.

Acknowledgements This Research was supported by the Sookmyung Women's University Research Grants (1-1603-2006).

References

1. Gubbi J, Buyyab R, Marusic S, Palaniswami M (2013) Internet of Things (IoT): a vision, architectural elements, and future directions. Future Gen Comput Syst 29:1645–1660
2. Shu C-F, Hampapur A, Lu M, Brown L, Connell J, Senior A, Tian Y (2005) IBM smart surveillance system (S3): a open and extensible framework for event based surveillance. In: IEEE Conference on Advanced Video and Signal Based Surveillance, pp 318–323
3. Cho Y, Lim SO, Yang HS (2010) Collaborative occupancy reasoning in visual sensor network for scalable smart video surveillance. IEEE Trans Consum Electron 56(3):1997–2003
4. Nazare AC, dos Santos CE, Ferreira R, Robson Schwartz W (2014) Smart surveillance framework: a versatile tool for video analysis. In: IEEE Winter Conference on Applications of Computer Vision (WACV), pp 753–760
5. Castro-Munoz G, Martinez-Carballido J (2015) Real time human action recognition using full and ultra high definition video. In: International Conference on Computational Science and Computational Intelligence (CSCI), pp 509–514
6. Bross B, Han W-J, Sullivan GJ, Ohm J-R, Wiegand T (July 2012) High efficiency video coding (HEVC) text specification draft 8, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC) document JCTVC-J1003
7. Sullivan GJ et al (2012) Overview of the high efficiency video coding (HEVC) standard. IEEE Trans Circuits Syst Video Technol 22(12):1649–1668
8. Il-Koo K et al (2012) Block partitioning structure in the HEVC standard. IEEE Trans Circuits Syst Video Technol 22(12):1697–1706
9. Chiang P-T, Chang TS (2013) Fast zero block detection and early CU termination for HEVC video coding. In: IEEE International Symposium on Circuits and Systems (ISCAS), pp 1640–1643
10. Pan Z, Kwong S, Zhang Y, Lei J (2014) Fast coding tree unit depth decision for high efficiency video coding. In: IEEE International Conference on Image Processing (ICIP), pp 3214–3218
11. Kibeya H, Belghith F, Ben Ayed MA, Masmoudi N (2014) A fast CU partitionning algorithm based on early detection of zero block quantified transform coefficients for HEVC standard. In: IEEE International Conference on Image Processing, Applications and Systems (IPAS), pp 1–5
12. Schierl T, Hannuksela MM, Wang Y-K, Wenger S (2012) System layer integration of high efficiency video coding (HEVC). IEEE Trans Circuits Syst Video Technol 22(12):1871–1884
13. Psannis KE, Hadjinicolaou M, Krikelis A (2006) MPEG-2 streaming of full interactive content. IEEE Trans Circuits Syst Video Technol 16(2):280–285
14. Wenger S (2003) H.264/AVC over IP. IEEE Trans Circuits Syst 13(7):645–656
15. Stockhamme T, Hannuksela MM, Wiegand T (2003) H.264/AVC in wireless environments. IEEE Trans Circuits Syst Video Technol 13(7):657–673
16. Psannis K, Ishibashi Y (2008) Efficient flexible macroblock ordering technique. IEICE Trans Commun E91–B(08):2692–2701
17. Psannis KE (2015) HEVC in wireless environments. J Real Time Image Process. doi:10.1007/s11554-015-0514-6 **(online published)**
18. Psannis K, Ishibashi Y (2006) Impact of video coding on delay and jitter in 3G wireless video multicast services. EURASIP J Wirel Commun Netw 2006, Article ID 24616, 1–7
19. Zhang Y, Wang H, Li Z (2013) Fast coding unit depth decision algorithm for inter-frame coding in HEVC. In: Proceedings of Data Compression Conference, pp 53–62
20. Shen L, Liu Z, Zhang X, Zhao W, Zhang Z (2013) An effective cu size decision method for HEVC encoders. IEEE Trans Multimed 15(2):465–470
21. Choi K, Park S-H, Jang ES (2011) Coding tree prunning based CU early termination. Document JCTVC-F092, JCT-VC
22. Yang J, Kim J, Won K, Lee H, Jeon B (2011) Early SKIP Detection for HEVC, document JCTVC-G543. JCV-VC, Geneva Switzerland
23. Kim J, Jeong S, Cho K, Choi JS (2012) An efficient bi-prediction algorithm for HEVC. In: International Conference on Consumer Electronics (ICCE), Las Vegas
24. Xiong J, Li H, Meng F, Zeng B, Zhu S, Wu Q (2014) Fast and efficient inter CU decision for high efficiency video coding In: Proceedings of IEEE International Conference Image Processing, pp 3715–3719
25. Xiong J, Li H, Zhu S, Wu Q, Zeng B (2014) MRF-based fast HEVC inter CU decision with the variance of absolute differences. IEEE Trans Multimed 16(8):2141–2153

26. Shen X, Yu L (2013) CU splitting early termination based on weighted SVM. EURASIP J Image Video Process 2013(1):1–11
27. Cho S, Kim M (2013) Fast CU splitting and pruning for suboptimal CU partitioning in HEVC intra coding. IEEE Trans Circuits Syst Video Technol 23(9):1555–1564
28. Xiong J, Li H, Wu Q, Meng F (2014) A fast HEVC inter CU selection method based on pyramid motion divergence. IEEE Trans Multimed 16(2):559–564
29. Lee J, Kim S, Lim K, Lee S (2015) A fast CU size decision algorithm for HEVC. IEEE Trans Circuits Syst Video Technol 25(3):411–421
30. Cassa M, Naccari M, Pereira F (2012) Fast rate distortion optimization for the emerging HEVC standard. In: Proceedings of Picture Coding Symposium (PCS), pp 493–496
31. Lee B, Kim M (2011) Modeling rates and distortions based on a mixture of laplacian distributions for inter-predicted residues in quadtree coding of HEVC. IEEE Signal Process Lett 18(10):571–574
32. Pan Z, Kwong S, Zhang Y, Lei J (2014) Fast coding tree unit depth decision for high efficiency video coding. In: IEEE International Conference of Image Processing (ICIP), pp 3214–3218
33. Li X, Wien M, Ohm JR (2010) Rate-complexity-distortion evaluation for hybrid video coding. In: IEEE International Conference Multimedia and Expo, Singapore, July 19–23
34. Bossen F (2012) Common test conditions and software reference configurations. Document JCTVC-I1100, JCT-VC, Geneva, Switzerland
35. Sampaio F, Bampi S, Grellert M, Agostini L, Mattos J (2012) Motion vectors merging: low complexity prediction unit decision heuristic for the inter-prediction of HEVC encoders. In: International Conference on Multimedia and Expo (ICME), pp 657–662
36. Yang S, Lee H, Shim HJ, Jeon B (2013) Fast inter mode decision process for HEVE encoder. In: The IEEE 11th Image, Video, and Multidimensional Signal Processing (IVMSP) Workshop, pp 1–4