J. Vis. Commun. Image R. 40 (2016) 34–41

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Computational complexity allocation and control for inter-coding of high efficiency video coding with fast coding unit split decision ☆

Jiunn-Tsair Fang [a], Zong-Yi Chen [b], Chang-Rui Lai [b], Pao-Chi Chang [b,*]

[a] Ming Chuan University, Department of Electronic Engineering, No. 5, Deming Rd., Taoyuan City 33348, Taiwan
[b] National Central University, Department of Communication Engineering, No. 300, Jhongda Rd., Taoyuan City 32001, Taiwan

ARTICLE INFO

Article history:
Received 19 January 2016
Revised 7 June 2016
Accepted 13 June 2016
Available online 15 June 2016

Keywords:
High-efficiency video coding (HEVC)
Complexity allocation
Complexity control
Coding unit
Motion vector

ABSTRACT

HEVC provides the quadtree structure of the coding unit (CU) with four coding-tree depths to facilitate high coding efficiency. However, compared with previous standards, the HEVC encoder increases computational complexity considerably, thus making it inappropriate for applications in power-constrained devices. This study therefore proposes a computational complexity allocation and control method for the low-delay P-frame configuration of the HEVC encoder. The complexity allocation includes the group of pictures (GOP) layer, the frame layer, and the CU layer in the HEVC encoder. Each layer involved uses individual method to distribute the complexity. In particular, motion vector estimation information is applied for CU complexity allocation and depth split determination. The total computational complexity can thus be reduced to 80% and 60% or even lower. Experiment results revealed that the average BD-PSNR exhibited a decrease of approximately 0.1 dB and a BD-bitrate increment of 2% when the target complexity was reduced to 60%.

1. Introduction

In recent years, mobile phones have become popular consumer electronics. Smartphones are ubiquitous in daily life for accessing the Internet for videos or multimedia applications. People are increasingly capturing and sharing multimedia on mobile devices; therefore, the demand for high-quality and real-time video is increasing. Hence, complex algorithms with heavy computations have been developed in recent video standards. Heavy computations, however, consume considerable power and reduce battery life, hindering video applications in power-constrained mobile devices. Therefore, reducing the computational complexity is crucial for video applications in mobile devices.

The video standard H.264/AVC employs the macroblock (MB) as the largest block for prediction coding [1]. An MB is a 16×16 block of pixels, and its prediction modes, partitioned from an MB with different block sizes, are used for motion estimation (ME). The optimal prediction mode is determined according to the rate-distortion optimisation (RDO) procedure. RDO reduces the prediction error but increases the computational complexity. Accordingly, mode decision (MD) or ME are popular topics for complexity reduction in H.264/AVC [2–5].

The newest video standard high-efficiency video coding, (HEVC), finalised in 2013, supports a coding efficiency that is higher than those of previous standards, especially for high-resolution video content [6]. HEVC uses quadtree structures of coding units (CUs) and four coding-tree depths to facilitate high coding efficiency. The largest coding unit (LCU) is composed of a block of 64×64 pixels, which is 16 times greater than the MB block size. The computational complexity is increased considerably, and many fast algorithms have been proposed based on CU depth decision in recent years [7–16].

Although fast algorithms for CU depth decision are intended to reduce the complexity of the HEVC encoder, the main objectives are to achieve a trade-off between the transmission quality and complexity. By contrast, the complexity control method is used for effectively maintaining the transmission quality as the target complexity is downscaled. The purpose of complexity control is not only to reduce the power consumed for a given target complexity but also to effectively control the transmission quality of a power-constrained device.

Extensive studies have been proposed to reduce complexity for H.264/AVC and HEVC, respectively. By contract, studies on complexity control were not as much. We list some representative work as follows. He et al. proposed a theoretical model for

☆ This paper has been recommended for acceptance by M.T. Sun.

* Corresponding author.
E-mail address: pcchang@ce.ncu.edu.tw (P.-C. Chang).

http://dx.doi.org/10.1016/j.jvcir.2016.06.004
1047-3203/

power-RD analysis. They adjusted complexity control parameters to match the available energy supply while maximizing the picture quality [17]. Chien et al. measured the coding gains of coding tools to reorder the steps of the encoding process. Some coding tools with less coding gains may be skipped to meet the required complexity constraint [18]. Corrêa et al. separated the sequence into constrained and unconstrained frames. Information obtained from the collocated areas in the previously encoded unconstrained frames could be used to predict the number of constrained frames and their coding-tree depths [19]. Several studies have investigated the improvement of the prediction performance [20–23]. Moreover, a workload management strategy was proposed for maintaining the frame rate transmission [24]. In [25], Zhang et al. proposed the motion collision count (MCC) to estimate the number of prediction units (PUs) located in the current encoding CU, and then determined the number of CU splits from each CU depth. The complexity control was based on the percentage of CU splits. Recently, a subjective driven complexity control was proposed to reduce and control the encoding complexity of HEVC [26].

This study focuses on the low-delay P-frame (LDP) configuration of HEVC, because this configuration is suitable for applications in low-power devices. The complexity allocation includes the group of pictures (GOP) layer, the frame layer, and the CU layer of the HEVC encoder. Instead of allocating exact computational complexity among frames or CUs, their relative complexity of consumption ratio was estimated. Different methods were applied to different coding layers to estimate the relative complexity consumption. As the target complexity was set, the relative complexity allocation for each coding layer could be preserved, and the complexity control could thus be applied effectively. To further increase the coding efficiency, a formula based on motion vector estimation was used for the fast decision of the CU depth split. Therefore, the proposed method could not only maintain the transmission quality adaptively but also and increase the coding efficiency of HEVC.

The remainder of this paper is organized as follows. Section 2 describes the proposed complexity control method, and the experimental results of are presented in Section 3. Finally, Section 4 presents the conclusion.

2. Complexity control method

In this study, the computational complexity was allocated to each coding layer of HEVC for complexity control. Methods for allocation are detailed in the following subsections. The computational complexity was measured by the operation of the central processing unit (CPU) clock cycles. The time consumption equals the consumed number clock cycles divided by the CPU frequency. First, the complexity was equally distributed among GOPs. Second, each frame complexity was estimated using a parabolic curve at different quantisation parameter (QP) settings. Third, the complexity distribution to each CU depended on the estimated number of PUs located in the current encoding CU block. Fourth, a formula was used for the fast decision of the LCU depth splitting. Finally, the computational complexity was compensated: Any over- or underallocated complexities were equally distributed among the remaining uncoded CUs or frames.

2.1. GOP layer complexity allocation

This study focused on the LDP configuration in the HEVC encoder [27]. The LDP configuration consisted of one I-frame, with the remainder being P-frames. Fig. 1 illustrates the LDP configuration, where each GOP contains four P-frames. The QP settings for the first and third frames were QP + 3, for the second frame, QP + 2,

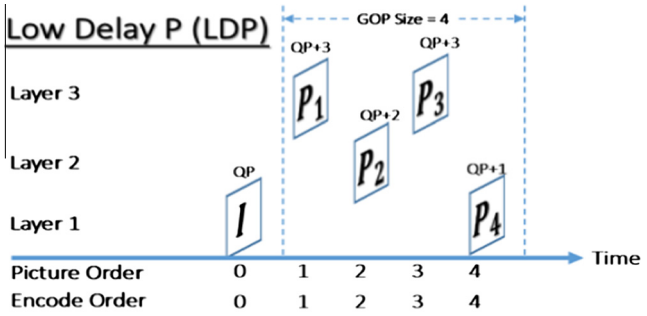

Fig. 1. Hierarchical coding structure of the LDP configuration [13].

and for the fourth frame, QP + 1. This QP setting was the same for the remaining GOPs. Fig. 1 also shows that the encoder order is the same as the picture order, meaning that this configuration was low-delay for real-time transmission [13].

To estimate the computational complexity in the GOP layer, four test sequences from different classes were simulated. The simulation environment is presented in Table 1. The reference software for HEVC was HM 12.1, and eight QP settings were selected. The computational complexity was determined according to the CPU clock. The sequences of Class D were excluded in this study because of their low resolution.

Fig. 2 shows the complexity consumption of the first 97 frames with QP 27 for four sequences (Classes A, B, C, and E). The complexity consumption depended on the frame size and sequence content. Large frame sizes with high-motion sequences, such as Class A_PeopleOnStreet, consumed more complexity than small frame sizes with low-motion sequences, such as Class E_Vidyol. Fig. 2 also shows a crucial characteristic of the LDP configuration: The complexity consumption for each sequence could be treated as a periodic signal with a period of 4, except for the first GOP. In other words, processing each GOP, composed of four frames, involved almost the same complexity.

Note that processing the first I frame involves a lower complexity compared with processing other frames. This is because P frame uses interprediction and I frame uses intraprediction for prediction coding. It usually takes more complexity to process interprediction than to process intraprediciton. From HEVC default setting, four reference frames are used for interprediction. The first four P frames (the first GOP) thus use less number of reference frames for interprediction, and involves a lower complexity compared with processing the other GOPs. For these four P frames, their complexity consumption increases as the number of available reference frames increases.

According to the LDP configuration, the complexity of all P frames, C_P, can be expressed as follows:

$$C_P = C_E - C_I \tag{1}$$

Table 1
Simulation environment.

Reference software	HM 12.1
Sequence	Class A_PeopleOnStreet Class B_Kimono1 Class C_RaceHorse Class E_Vidyo1
FramesToBeEncoded	97
Configuration file	Low Delay P (IPPP...)
QP	20, 22, 25, 27, 30, 32, 35, 37, 40, 42
Hardware	Intel(R) Core(TM) i7 CPU 920 @ 2.67 GHz,4.0 GB of RAM

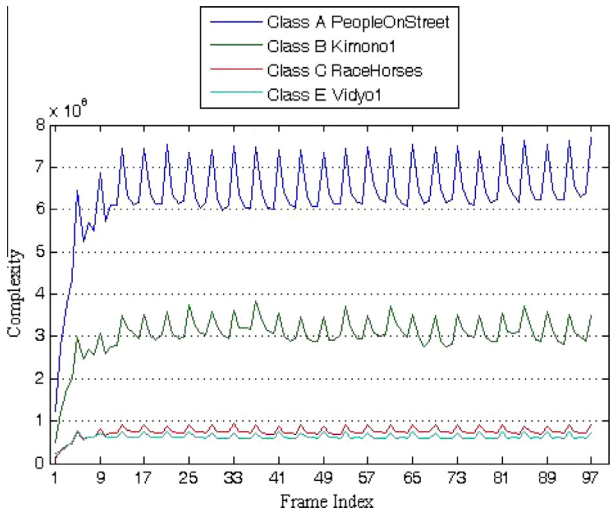

Fig. 2. Frame complexity consumption from four test video sequences.

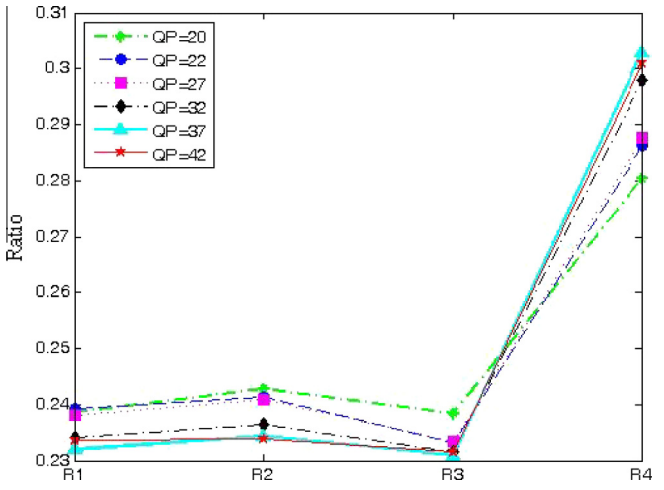

Fig. 3. Average ratio of frame complexity consumption for each GOP under different QP settings.

where C_E is the entire complexity of the sequence, and C_I is the complexity of the I frame. The complexity for I frame can be estimated according to the average complexity of I frame from those eight test sequences. From the result of Fig. 2, the complexity allocation to each GOP, C_{GOP}, can be expressed as follows,

$$C_{GOP} = C_P / N_{GOP} \tag{2}$$

where N_{GOP} is the number of GOPs in the sequence; that is, the complexity for each GOP was equally distributed.

2.2. Frame layer complexity allocation

The allocation of the frame complexity at different QP settings was determined using a parabolic curve. As Fig. 2 illustrates, the complexity of each GOP was almost identical, and the frame-encoding time in the same position for each GOP had similar complexity. According to these results, the frame complexity estimation could be simplified only to estimate the complexity of each frame in a GOP, because frames in the same position of each GOP had similar complexity consumption. According to the LDP configuration, four frames composed a GOP; thus, only the complexity of four frames had to be estimated.

Let R_i denote the complexity ratio of the ith position frame to the GOP, where the value of i is 1, 2, 3, or 4. The R_i for each position is calculated according to the test sequences with different QP settings; the results are plotted in Fig. 3. It only plots six different QP setting for clearer representation. For low QPs, the effect of the complexity consumption on the QP factor for each frame is substantial. However, for high QPs, the rates of the first three frames are similar, and the rate of the fourth frame is highest.

A parabolic curve was applied to fit each R_i for different QPs. The estimated ratio for the ith frame, denoted as $\hat{R}_i$, starting from QP value at 20, can be expressed according to the relation (3),

$$\widehat{R}_i = a_i \times QP^2 + b_i \times QP + c_i \tag{3}$$

where a_i, b_i, and c_i are constants and the value of i is 1, 2, 3, or 4. Table 2 lists the values of these constants, and Fig. 4 shows each R_i and its estimated curve. The estimated error of each R_i is less than 1.25%. Finally, the complexity of the P frame at the ith position of a GOP, C_{F_i}, is estimated according to the expression

$$C_{F_i} = \hat{R}_i \times C_{GOP} \tag{4}$$

Table 2
Constant parameters for frame complexity.

i	a_i	b_i	c_i
1	5×10^{-5}	-1.2×10^{-3}	2.416×10^{-1}
2	2×10^{-5}	-9.7×10^{-4}	2.425×10^{-1}
3	5×10^{-5}	-1.1×10^{-3}	2.369×10^{-1}
4	1×10^{-4}	3.3×10^{-3}	2.789×10^{-1}

2.3. CU layer complexity allocation

The complexity allocation in the CU layer was to distribute the complexity from the frame layer to the CU layer, including the LCU layer, and CU in Depth 1. Similar to the frame layer allocation, the complexity allocation to the CU layer involves estimating the relative complexity consumption.

For the LCU layer, to efficiently allocate the complexity among LCUs within a frame, a metric is necessary to estimate the computational complexity of each LCU. The HEVC rate control algorithm employs a linear model to estimate the mean absolute difference (MAD) of the current encoding frame by using the MAD of its previous frame, because there is a strong correlation between the successive frames [28]. In this study, the concept of MCC, which refers to the motion information of the PU in the previous frame, was adopted to estimate the complexity of each LCU [25]. Zhang et al. recorded all the MVs and related partitions of PUs of the previous frame; in their study, they allowed the PUs to move forward at the same distance but in the opposite direction to their MVs. They then counted the number of PUs located in each LCU block of the current encoding frame. This number was the MCC. Fig. 5 shows the simulation results indicating the MCC location, plotting by blue dots, within each LCU from the second frame of the sequence Class C_RaceHorse. The LCU block containing high-motion or complex content generally had more MCC numbers than did the LCU block containing low-motion or homogeneous content. Therefore, the concept of the MCC was used for complexity allocation to the CU layer; that is, the complexity for the current encoding LCU was proportional to its MCC value. The complexity allocation ratio to the jth LCU in the frame, denoted as R_{LCU_j}, is estimated using

$$R_{LCU_j} = \frac{MCC_j}{\sum_{j=1}^{N_{LCU}} MCC_j} \tag{5}$$

where MCC_j is the MCC value of the jth LCU, and N_{LCU} is the total number of LCUs in the frame.

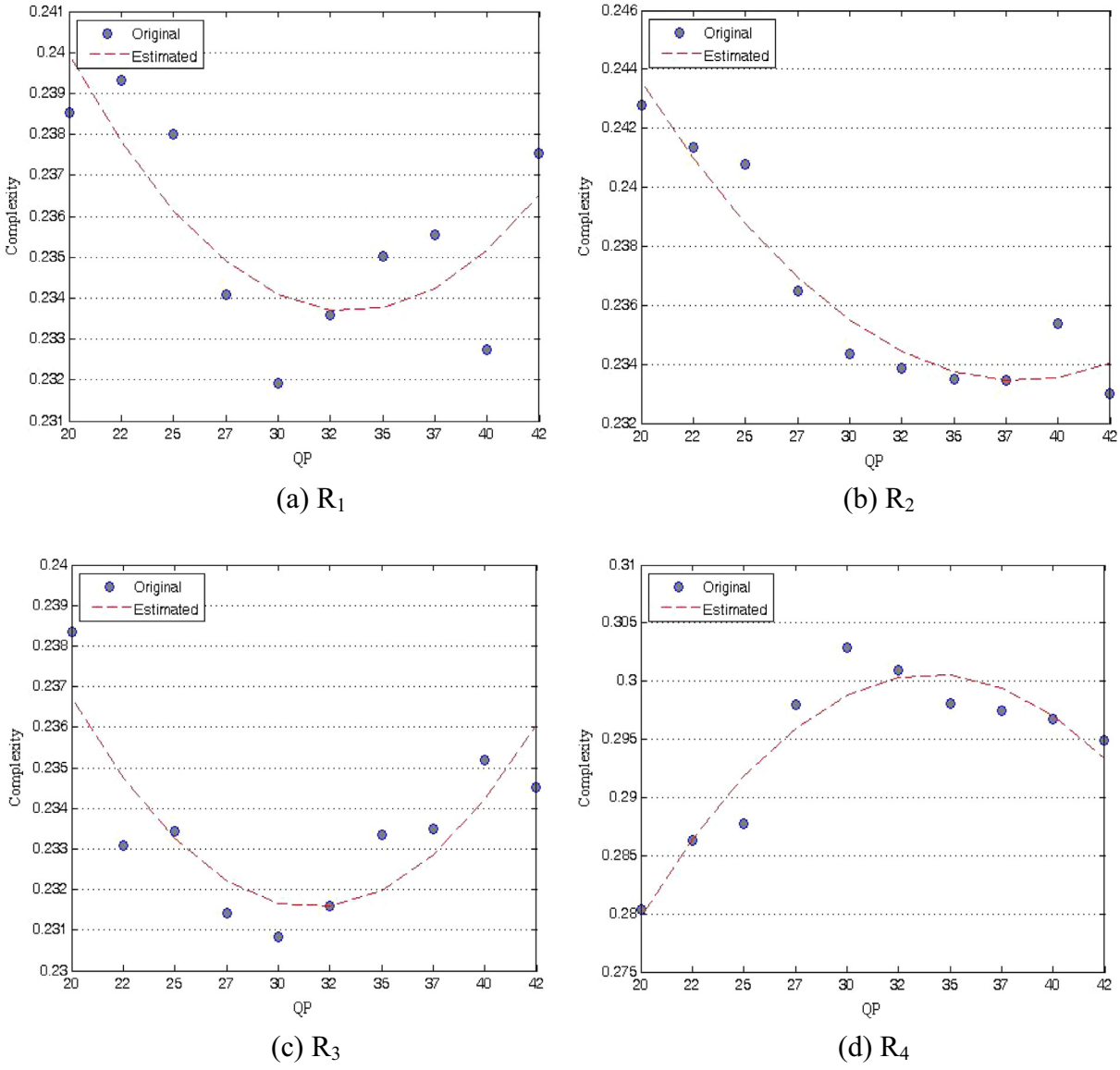

Fig. 4. Estimated and original average ratio for frame complexity under different QP settings.

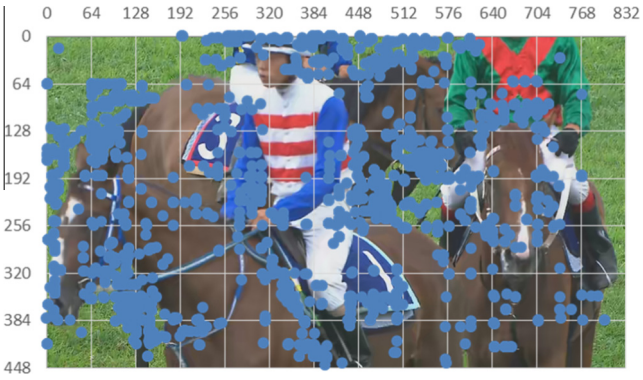

Fig. 5. An example showing the MCC location in each LCU block.

This method can also be applied for the complexity allocation of other CU coding depths. Because an LCU has four CUs in Depth 1, the complexity from each LCU is distributed only to its four CUs. The complexity allocation to CUs in Depth 1 is proportional to the MCC number located in each CU block. In other words, a CU with a large MCC has more complexity. By contrast, a CU with a small MCC may quickly terminate its encoding process. Because the coding efficiency of CU in Depth 2 or 3 is less than the CU in Depth 0 or 1, the complexity of the CU in Depths 2 and 3 are not controlled using the proposed method.

2.4. Fast algorithm for the LCU split decision

HEVC provides CU a quadtree structure, and uses the RDO procedure to determine the best PU for each CU. Though the RDO procedure can improve coding efficiency, it increases computational complexity considerably. Therefore, in this work, in addition to allocating the complexity to each coding layer for complexity control, a fast algorithm for CU split decision is required to reduce the computational complexity for HEVC encoder.

The MCC value of each test sequence was measured. Fig. 6 plots the relationship between the MCC average and QPs, where the MCC average in a frame was the total MCC number divided by the LCU number. The graph shows that high-motion sequences generally had higher MCC average values than low-motion

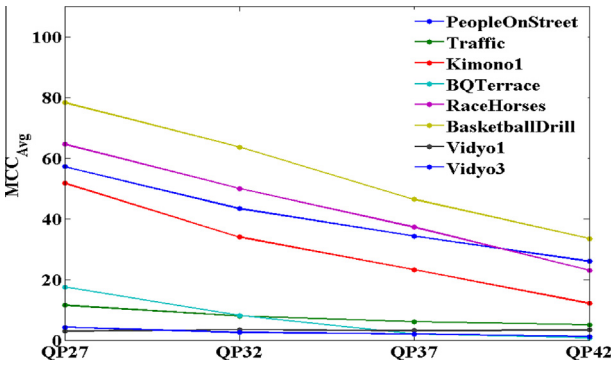

Fig. 6. Relation between average MCC and QPs for different test sequences.

sequences. This is because high-motion sequences required more split PUs than low-motion sequences for interprediction. Moreover, the average MCC increased as QP decreased, as more content details can be preserved from a small QP than from a large QP for interprediction. In other words, the MCC value reflected the sequence content and QP setting for interprediction. Therefore, we assume that a frame with a high MCC average has a high ratio of LCU split. Conversely, a frame with a low MCC average has a low ratio of LCU split.

About the LCU depth split estimation, the ratio of LCU depth split, R_{split}, is estimated by the average MCC. It can be derived as $R_{split} = f(MCC_{avg})$ where f is a function operator, and MCC_{avg} is a parameter. Fig. 7 shows that the relation between R_{split} and MCC_{avg} based on two factors, video content (VC) and QP. From Fig. 7, if the observation is based on the same video sequence, it shows that the R_{split} decreases as QP increases, and the R_{split} increases as QP decreases. Furthermore, MCC_{avg} is inversely proportional to QP for a specific video sequence. If the observation is based on the same QP, it shows high motion sequence has high R_{split}, and low motion sequence has low R_{split}. Furthermore, MCC_{avg} is positive proportional relation to VC for a given QP.

Combining these two factors, QP and VC, a quadratic form is used to model the relation between R_{split} and MCC_{avg}, expressed by (6),

$$R_{split} = -t_1 \times MCC_{avg}^2 + t_2 \times MCC_{avg} + t_3 \tag{6}$$

where t_1, t_2, and t_3 are constant. These coefficients can be obtained by simulating the test sequences.

The MCC value of each LCU in each frame needs to be sorted, just the same procedure as [25] did. With the sorting result and

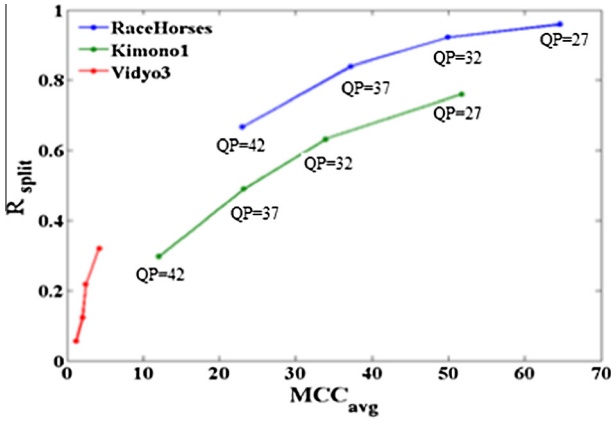

Fig. 7. Relation between LCU-depth split ratio and average MCC based on video sequences and QPs.

(6), the LCU depth split can be estimated. Compared with the method in [25], in the proposed method, CUs with higher MCC values are also with a higher chance to be split, but the LCU split ratio is determined by (6).

Table 3 lists the correctness of the LCU split decision under the high motion sequence (RaceHorse) and the low motion sequence (Vidyo 1). The average correctness is over 90%, which means (6) is quite applicable. The result is similar to the result in Table 1 [25].

Formula (6) can also be applied to determine the split of CUs in Depth 1. Because the size of the CUs in Depth 1 is only one quarter the size of the LCU, the average MCC value is only divided by 4, and the threshold of CUs in Depth 1 is one-quarter of the threshold, T, of the LCU in (6). The proposed method was not applied for the fast algorithm for the CUs in Depths 2 and 3.

2.5. Complexity compensation

The actual complexity consumption may not be equivalent to the allocated complexity. First, some LCU or CU early terminates its process (or using skip mode), the extra distributed complexity can thus be distributed to the rest frames or CUs. Second, as the distributed complexity is exhausted, but the process is still going, this procedure can continue until it ends. However, the extra used complexity will be deduced from the rest frames or CUs.

As above description, when a frame consumes a complexity different from that which is allocated, the complexity allocation to the next frame is adjusted by adding the complexity compensation C_{CMP}, which is calculated using the expression

$$C_{CMP} = (A_{F_i} - C_{F_i})/(n - n_i) \tag{7}$$

where A_{F_i} is the actual complexity consumption of the ith frame F_i, n is the total number of frames in the GOP, n_i is the ith frame, and $n - n_i$ represents the remaining uncoded frames in this GOP.

The complexity offset $A_{F_i} - C_{F_i}$ is used to adjust the complexity allocation. If the allocated complexity is more than the actual complexity consumption, then this term is negative, and the complexity allocation for the remaining frames is reduced. Conversely, when this term is positive, the complexity allocation for the remaining frames is increased. Finally, this complexity offset is divided among the remaining uncoded frames for equal distribution.

The formula for the complexity compensation of the LCU layer is similar to (7). The complexity offset term in the numerator is replaced with the LCU complexity offset, and the term representing the remaining uncoded frames in the denominator is replaced by the remaining uncoded LCUs.

Fig. 8 shows the block diagram of the proposed method. As the total complexity is determined, the complexity is equally distributed to each GOP. The complexity distribution to each GOP is based on (1) and (2). Formula (3) estimates the ratio of complexity consumption for each frame under a given QP, and the complexity allocation from a GOP to each of its fours frames is based on (4). The complexity allocation for each LCU is proportional to its MCC value, and (5) estimates the ratio of complexity consumption for

Table 3
LCU-depth splitting based on the proposed method.

RaceHorse		Vidyo1	
QP	Correctness	QP	Correctness
27	0.98	27	0.96
32	0.96	32	0.98
37	0.93	37	0.90
42	0.94	42	0.90
Average	0.95	Average	0.93

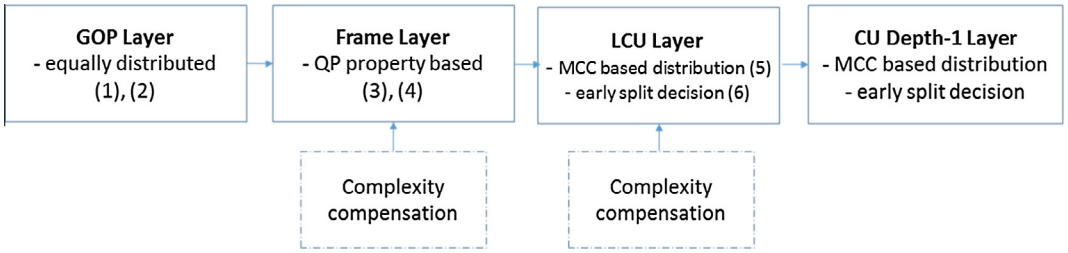

Fig. 8. A block diagram for proposed method.

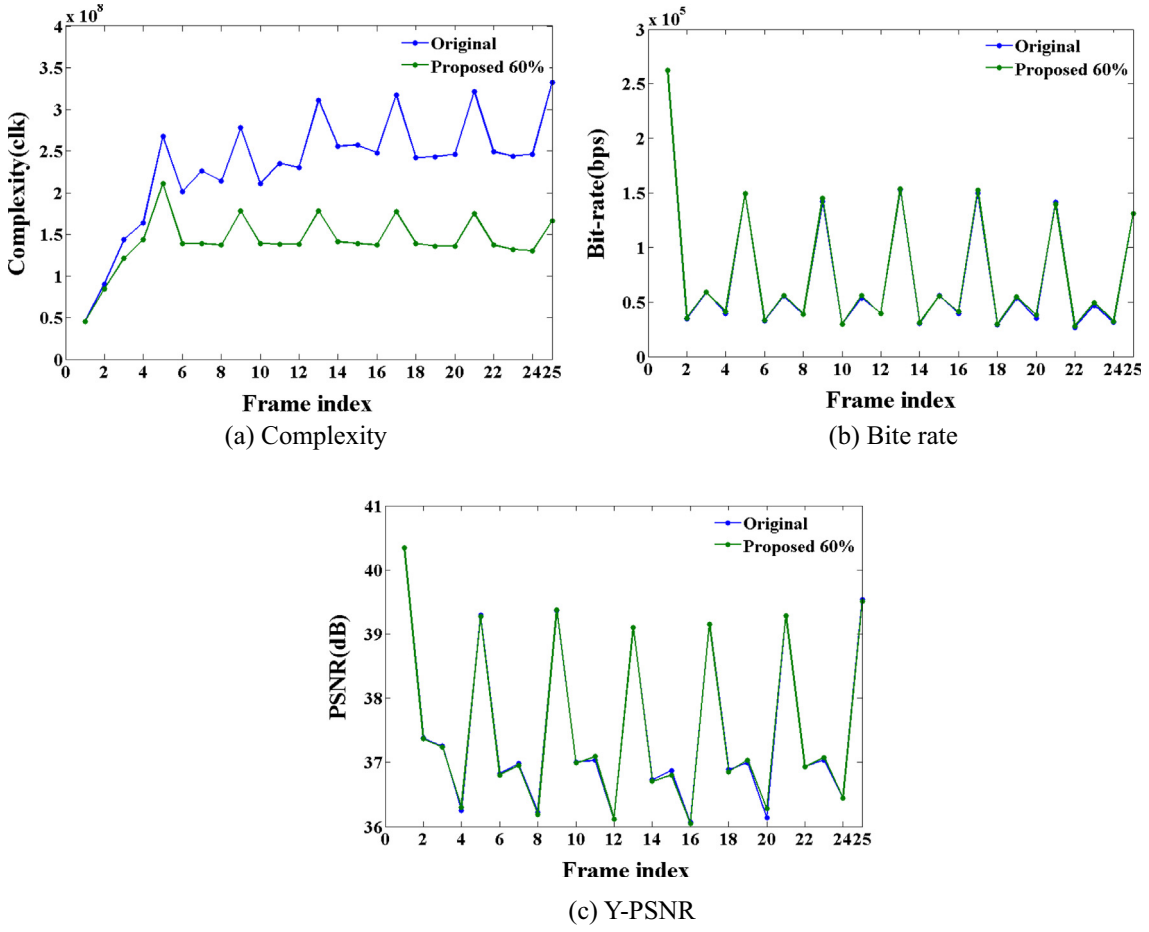

Fig. 9. Performance of Class C_RaceHorses from the first 30 frames under a QP of 32 and 60% of target complexity.

each LCU. To improve coding efficiency, (6) shows the relation between average MCC and LCU split ratio. A fast algorithm to determine the LCU depth split is thus proposed. Similar methods for complexity allocation and depth split algorithm are applied to CU Depth 1. Finally, the scheme of complexity compensation is derived for frame layer or LCU layer, respectively.

In this work, the complexity allocation includes the group of pictures (GOP) layer, the frame layer, and the CU layer in the HEVC encoder. Each layer involved uses individual method to distribute the complexity based on their relative ratios among its sub-layers. As the target complexity is changed, these relative ratios for complexity allocation are still preserved. This is the main strategy of the proposed method. The complexity of HEVC encoder can thus be controlled.

3. Experiment results

The simulations were designed to show the RD performance with the target complexity set at 80%, 60%, or 40%. Simulation environment was the same as listed in Table 1, and four test sequences were added, Class A_Traffic, Class B_BQTerrace, Class C_BasketballDrill, and Class E_Vidyo3. In practice, for real time complexity control, the complexity allocation of each GOP can be estimated by its previous encoded GOPs. However, to accurately evaluate the system performance, Corrêa et al. did not set up the experiment for real time. The total computational complexity available for encoding (i.e., 100%) was determined beforehand by encoding each sequence without imposing any complexity constraint. The average result was defined as the maximum available complexity

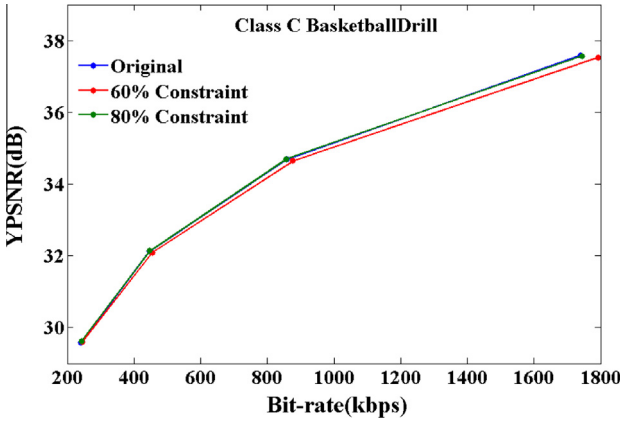

Fig. 10. RD performance of the Class C BasketballDrill at 80% and 60% of target complexity.

available for encoding a sequence. The target complexity expressed as percentages of total available complexity was then defined to evaluate the performance of the proposed method [19].

To evaluate the error between the target complexity and the actual consumed complexity, the complexity control error (CCE) was defined as follows:

$$CCE\ (\%) = [(C_{pro} - C_T)/C_T] \times 100 \tag{8}$$

where C_{pro} is the actual consumed complexity determined using the proposed method, and C_T is the target complexity. The conditions of the simulation environment are listed in Table 1. Only four QP settings were used: 27, 32, 37, and 42. The constant numbers, t_1, t_2, and t_3, in (6) were simulated to be 0.01, 1, and 0, respectively.

The first simulation was conducted to test the RD performance. A test sequence with a QP of 32 (Class C_RaceHorse) was selected as an example to show the complexity consumption and the RD performance of each frame. Fig. 9(a)–(c) illustrates the complexity, bit-rate, and Y-PSNR performance for the first 25 frames. The complexity consumption of each frame appeared to have decreased compared with the original complexity (Fig. 9(a)). The fourth frame in each GOP still consumed the largest complexity. Fig. 9(b) and (c) shows that the bit-rate and Y-PSNR performance approximated the performance of the original sequence for each frame. Because the fourth frame in each GOP was allocated more bit-rates than the other three frames, its Y-PSNR performance was closer to the original performance compared with the Y-PSNR performance of the other three frames.

Fig. 10 shows the RD performance for the same test sequence. The RD performance decreased slightly as the percentage of the target complexity decreased from 80% to 60%. Tables 4 and 5 list the Bjøntegaard delta rate (BD-rate), BD-PSNR [29], and CCE for the target complexities of 80% and 60%. The sequence of Class B_Kimono showed less deterioration in the BD-rate and BD-PSNR performance compared with the other sequences. This is because Class B_Kimono1 is a low-motion sequence; the early termination in its prediction process (from the limited complexity) did not affect the performance appreciably. Conversely, the drop in the BD-rate and BD-PSNR in high-motion sequences, such as Class A_Traffic and Class C_BasketballDrill, was larger than that of the other sequences. As the target complexity dropped from 80%, the BD-rate increased from 0.11%, the BD-PSNR affected little, and the CCE was approximately 0.42%. However, as the target complexity dropped to 60%, the BD-rate increased from 2.1%, the BD-PSNR

Table 6
Average BD-rate, BD-PSNR, and CCE performance under 40% of target complexity.

	BD-Bitrate (%)	BD-YPSNR (dB)	CCE (%)
Traffic	6.44	−0.236	1.40
Kimono1	0.015	−0.058	2.02
BQTerrace	2.66	−0.071	1.07
Vidyo3	4.25	−0.176	1.10
Average	3.34	−0.135	1.39

Table 4
Average BD-rate, BD-PSNR, and CCE performance under 80% of target complexity.

	BD-Bitrate (%)		BD-PSNR (dB)		CCE (%)	
	Zhang [25]	Proposed	Zhang [25]	Proposed	Zhang [25]	Proposed
PeopleOnStreet	0.529	−0.277	−0.026	0.014	−0.85	−0.67
Traffic	1.65	0.84	−0.063	−0.032	0.54	0.47
Kimono1	0.005	−0.005	−0.019	0.019	0.47	0.43
BQTerrace	0.733	0.049	−0.019	−0.002	0.46	0.40
BasketballDrill	0.709	−0.187	−0.029	0.007	0.37	0.30
RaceHorse	1.906	0.180	−0.067	0.000	−0.60	−0.44
Vidyo1	0.480	0.267	−0.02	−0.01	0.40	0.35
Vidyo3	0.685	0.026	−0.029	0.001	0.43	0.36
Average	0.83	0.111	−0.034	0.00038	0.52	0.42

Table 5
Average BD-rate, BD-PSNR, and CCE performance under 60% of target complexity.

	BD-Bitrate (%)		BD-PSNR (dB)		CCE (%)	
	Zhang [25]	Proposed	Zhang [25]	Proposed	Zhang [25]	Proposed
PeopleOnStreet	5.525	4.195	−0.258	−0.198	0.54	0.47
Traffic	4.67	2.54	−0.173	−0.095	0.64	0.60
Kimono1	0.015	0.003	−0.058	−0.013	0.56	0.56
BQTerrace	1.828	1.024	−0.049	−0.028	0.58	0.55
BasketballDrill	5.187	2.977	−0.202	−0.118	0.30	0.23
RaceHorse	7.163	4.631	−0.259	−0.169	−0.43	−0.36
Vidyo1	1.248	0.841	−0.036	−0.036	0.43	0.39
Vidyo3	2.564	0.800	−0.109	−0.032	0.60	0.57
Average	3.52	2.126	−0.145	−0.086	0.51	0.43

increased from 0.086 dB, and CCE was approximately 0.43%. Finally, the proposed method demonstrated superior BD-rate, BD-PSNR, and CCE performance to that of the method proposed in [25], regardless of whether the target complexity dropped from 80% or 60%.

Zhang et. al. propose MCC to estimate CU depth split [25]. In their proposed method, CUs with higher MCC values are with a higher chance to be split, but the ratio of CU depth split is determined by the available complexity. In this work, the parameter MCC is adopted. The difference between the proposed method and [25] is that we separated the CU depth split and CU complexity allocation problems and solved them by two different methods.

Finally, Table 6 lists the BD-bitrate, BD-PSNR, and CCE for the target complexities of 40%. It shows that the proposed method still maintains acceptable performance.

4. Conclusion

This study proposes a complexity control method for LDP configuration of HEVC encoder. Complexity was allocated for the GOP layer, frame layer, and LCU layer to the CU depths. The complexity of the HEVC encoder is effectively controlled by the proposed method of layer-by-layer complexity allocation. In particular, motion vector estimation information was applied for CU complexity allocation and depth split determination. Experiment results shows that the proposed method could maintain acceptable BD-bitrate and BD-PSNR as the complexity reduces to 80%, 60%, or even 40%. Also, the complexity control error was little. The proposed method improves HEVC coding efficiency and adequacy for video applications in power-constrained devices.

References

[1] ITU-T Rec. H.264 and ISO/IEC 14496-10 (MPEG-4 AVC), ITU-T and ISO/IEC JTC 1, Advanced Video Coding for Generic Audiovisual Services, Version 8 (Including SVC Extension), 2007.

[2] Z. Li, G. Wen, Reusable architecture and complexity-controllable algorithm for the integer/fractional motion estimation of H.264, IEEE Trans. Consum. Electron. 53 (2) (2007) 749–756.

[3] S. Saponara, M. Casula, F. Rovati, D. Alfonso, L. Fanucci, Dynamic control of motion estimation search parameters for low complex H.264 video coding, IEEE Trans. Consum. Electron. 52 (1) (2006) 232–239.

[4] J. Ren, N. Kehtarnavaz, M. Budagavi, Computationally efficient mode selection in H.264/AVC video coding, IEEE Trans. Consum. Electron. 54 (2) (2008) 877–886.

[5] D. Martinez-Enriquez, A. Jimenez-Moreno, F. Diaz-de-Maria, An adaptive algorithm for fast inter mode decision in the H.264/AVC video coding standard, IEEE Trans. Consum. Electron. 56 (2) (2010) 826–834.

[6] G.J. Sullivan, J.R. Ohm, W.J. Han, T. Wiegand, Overview of the high efficiency video coding (HEVC) standard, IEEE Trans. Circ. Syst. Video Technol. 22 (12) (2012) 1649–1668.

[7] T. Zhao, Z. Wang, S. Kwong, Flexible mode selection and complexity allocation in HEVC efficiency video coding, IEEE J. Select. Top. Signal Process. 7 (6) (2013) 1135–1144.

[8] J. Vanne, M. Viitanen, T.D. Hämäläinen, Efficient mode decision schemes for HEVC inter prediction, IEEE Trans. Circ. Syst. Video Technol. 24 (9) (2014) 1579–1593.

[9] J. Lee, S. Kim, K. Lim, S. Lee, A fast CU size decision algorithm for HEVC, IEEE Trans. Circ. Syst. Video Technol. 25 (3) (2015) 411–421.

[10] J. Xiong, H. Li, Q. Wu, F. Meng, A fast HEVC inter CU selection method based on pyramid motion divergence, IEEE Trans. Multimedia 16 (2) (2014) 559–564.

[11] L. Shen, Z. Liu, X. Zhang, W. Zhao, Z. Zhang, An effective CU size decision method for HEVC encoders, IEEE Trans. Multimedia 15 (2) (2013) 465–470.

[12] S. Ahn, B. Lee, M. Kim, A novel fast CU encoding scheme based on spatiotemporal encoding parameters for HEVC inter coding, IEEE Trans. Circ. Syst. Video Technol. 25 (3) (2015) 422–435.

[13] J. Vanne, M. Viitanen, T.D. Hämäläinen, A. Hallapuro, Comparative rate-distortion-complexity analysis of HEVC and AVC video codecs, IEEE Trans. Circ. Syst. Video Technol. 22 (2012) 1885–1898.

[14] H. Lee, H.J. Shim, Y. Park, B. Jeon, Early skip mode decision for HEVC encoder with emphasis on coding quality, IEEE Trans. Broadcast. 61 (3) (2015) 388–397.

[15] L. Shen, Z. Zhang, Z. Liu Z, Adaptive inter-mode decision for HEVC jointly utilizing inter-level and spatiotemporal correlations, IEEE Trans. Circ. Syst. Video Technol. 24 (10) (2014) 1709–1722.

[16] G. Tian, X. Jin X, S. Goto, Content adaptive hierarchical decision of variable coding block sizes in high efficiency video coding for high resolution videos, IEICE Trans. Fundam. E96-A (4) (2013) 780–789.

[17] Z. He, Y. Liang, L. Chen, I. Ahmad, D. Wu, Power-rate-distortion analysis for wireless video communication under energy constraints, IEEE Trans. Circ. Syst. Video Technol. 15 (5) (2005) 645–658.

[18] M.C. Chien, Z.Y. Chen, P.C. Chang, Coding-gain-based complexity control for H.264 video encoder, in: Proc. IEEE Int. Conf. Image Processing (ICIP), California, USA, 2008, pp. 2136–2139.

[19] G. Corrêa, P. Assuncao, L. Agostini, L.A. da Silva Cruz, Complexity control of high efficiency video encoders for power-constrained devices, IEEE Trans. Consum. Electron. 57 (4) (2011) 1866–1874.

[20] G. Corrêa, P. Assuncao, L. Agostini, L.A. da Silva Cruz, Adaptive coding tree for complexity control of high efficiency video encoders, in: Proc. Picture Coding Symposium (PCS), Krakow, Poland, 2012, pp. 425–428.

[21] G. Corrêa, P. Assuncao, L. Agostini, L.A. da Silva Cruz, Coding tree depth estimation for complexity reduction of HEVC, in: Proc. Data Compression Conference (DCC), Snowbird, UT, 2013, pp. 43–52.

[22] G. Corrêa, P. Assuncao, L. Agostini, L.A. da Silva Cruz, Computational complexity control for HEVC based on coding tree spatio-temporal correlation, in: Proc. IEEE 20th International Conference on Electronics, Circuits, and Systems (ICECS), Abu Dhabi, UAE, 2012, pp. 937–940.

[23] G. Corrêa, P. Assuncao, L. Agostini, L.A. da Silva Cruz, Dynamic tree-depth adjustment for low power HEVC encoders, in: Proc. IEEE Int. Conf. Electronics, Circuits and Systems (ICECS), Seville, Spain, 2012, pp. 564–567.

[24] M. Grellert, M. Shafique, M.U.K. Khan, L. Agostini, J.C.B. Mattos, L.J. Henke, An adaptive workload management scheme for HEVC, in: Proc. IEEE Int. Conf. Image Processing (ICIP), Melbourne, 2013, pp. 1850–1854.

[25] Y. Zhang, S. Huang, H. Li, H. Chao, An optimally complexity scalable multi-mode decision algorithm for HEVC, in: Proc. IEEE Int. Conf. Image Processing (ICIP), Melbourne, VIC, 2013, pp. 2000–2004.

[26] X. Deng, M. Xu, L. Jiang, Subjective-driven complexity control approach for HEVC, IEEE Trans. Circ. Syst. Video Technol. 26 (1) (2016) 91–106.

[27] JCT-VC: High Efficiency Video Coding (HEVC) Test Model 12 (HM12) Encoder Description, JCTVC-N1002, 14th JCTVC Meeting, Vienna, AT, 2013.

[28] H. Choi, J. Nam, J. Yoo, D. Sim, I.V. Bajić, Rate Control Based on Unified RQ Model for HEVC, ITU-T SG16 Contribution, JCTVC-H0213, San José, 2012.

[29] G. Bjøntegaard, Calculation of Average PSNR Differences between RD-Curves, ITU-T SG16 Q.6 Document, VCEG-M33, Austin, US, 2001.