Journal of Electronic Imaging

JElectronicImaging.org

Complexity control for high-efficiency video coding by coding layers complexity allocations

Jiunn-Tsair Fang
Kai-Wen Liang
Zong-Yi Chen
Wei Hsieh
Pao-Chi Chang

SPIE. IS&T imaging.org

Jiunn-Tsair Fang, Kai-Wen Liang, Zong-Yi Chen, Wei Hsieh, Pao-Chi Chang, "Complexity control for high-efficiency video coding by coding layers complexity allocations," *J. Electron. Imaging* **25**(2), 023024 (2016), doi: 10.1117/1.JEI.25.2.023024.

Journal of Electronic Imaging 25(2), 023024 (Mar/Apr 2016)

Complexity control for high-efficiency video coding by coding layers complexity allocations

Jiunn-Tsair Fang,[a] Kai-Wen Liang,[b] Zong-Yi Chen,[b] Wei Hsieh,[b] and Pao-Chi Chang[b,*]
[a]Ming Chuan University, Department of Electronic Engineering, No. 5, Deming Road, Taoyuan City 33348, Taiwan
[b]National Central University, Department of Communication Engineering, No. 300, Jhongda Road, Taoyuan City 32001, Taiwan

Abstract. The latest video compression standard, high-efficiency video coding (HEVC), provides quad-tree structures of coding units (CUs) and four coding tree depths to facilitate coding efficiency. The HEVC encoder considerably increases the computational complexity to levels inappropriate for video applications of power-constrained devices. This work, therefore, proposes a complexity control method for the low-delay *P*-frame configuration of the HEVC encoder. The complexity control mechanism is among the group of pictures layer, frame layer, and CU layer, and each coding layer provides a distinct method for complexity allocation. Furthermore, the steps in the prediction unit encoding procedure are reordered. By allocating the complexity to each coding layer of HEVC, the proposed method can simultaneously satisfy the entire complexity constraint (ECC) for entire sequence encoding and the instant complexity constraint (ICC) for each frame during real-time encoding. Experimental results showed that as the target complexity under both the ECC and ICC was reduced to 80% and 60%, respectively, the decrease in the average Bjøntegaard delta peak signal-to-noise ratio was ~0.1 dB with an increase of 1.9% in the Bjøntegaard delta rate, and the complexity control error was ~4.3% under the ECC and 4.3% under the ICC. [DOI: 10.1117/1.JEI.25.2.023024]

Keywords: high-efficiency video coding; complexity allocation; complexity control; coding unit.

Paper 15768 received Oct. 20, 2015; accepted for publication Mar. 25, 2016; published online Apr. 19, 2016.

1 Introduction

Mobile phones have become popular consumer electronic devices in recent years. Accessing the internet for multimedia communications or video applications by using smartphones is ubiquitous in daily life. Gradually, people are becoming accustomed to capturing or sharing multimedia on mobile devices, therefore, the demand for high-quality and real-time video is increasing. For high-quality real-time video transmission, complex algorithms involving a large amount of computations have been developed in recent video coding standards. Heavy computation consumes considerable power, reducing battery life, and hinders video applications on power-constrained mobile devices. Therefore, reducing the computational complexity is crucial for video applications on mobile devices and these results in a complexity control mechanism being involved in video codec. The purpose of the complexity control is not only to reduce the power consumption for a given target complexity, but also to effectively control the transmission quality on a power-constrained device.

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 some different sizes of blocks, are used for motion estimation (ME). The optimal prediction mode is determined according to the rate-distortion optimization (RDO) procedure. RDO reduces the prediction error, but increases the computational complexity. Thus, fast algorithms for mode decision (MD) or ME are popular topics for complexity control in H.264/AVC.

The newest video standard high-efficiency video coding (HEVC) was finalized in 2013.[2] HEVC employs quad-tree structures of coding units (CUs) and four coding tree depths to facilitate high coding efficiency, especially for high-resolution video content. However, the computational complexity in HEVC encoder is considerably increasing. Corrêa et al.[3] indicated that the coding efficiency of HEVC would be decreased at some point in the encoding process, even though more computational complexity was involved. They further proposed that the maximal HEVC complexity could be reduced by balancing the trade-off between complexity and coding efficiency.

Accordingly, in this study, two scenarios of complexity constraint were considered for real-time video transmission on a mobile device: the entire complexity constraint (ECC) and the instant complexity constraint (ICC). The ECC is the entire computational complexity constraint for the video sequence to be encoded. If the total complexity consumption of a sequence is constrained to the ECC, then the total power consumption for encoding this sequence can be confined to a target complexity. The ICC for real-time video coding is that the allowable complexity for encoding each frame is the same and is limited so that the coding time of each frame can be controlled.

The complexity control methods based on the ECC condition for H.264 or HEVC can be referred to as follows: the parameters of fraction ME and integer ME were reused to meet system requirements.[4] A complexity control method was achieved to match the MD threshold to a target complexity level.[5] Lee et al.[6] proposed a fast reference frame selection algorithm based on information from the reference region for

*Address all correspondence to: Pao-Chi Chang, E-mail: pcchang@ce.ncu.edu.tw

an ME process. A content-adaptive MD based on the RD cost statistics was proposed for reducing the encoding time.[7] Chien et al.[8] reordered the steps of the encoding process by measuring the coding gains of coding tools. In Ref. 9, an operational method was proposed for RD and complexity optimization. He et al.[10] first modeled the coding tools as parameters and then proposed a theoretical model for power-RD analysis.

Corrêa et al. separated the sequence into unconstrained frames and constrained frames. Information obtained from the colocated area in the previous encoded unconstrained frame could be used to predict the number of constrained frames and their coding tree depths.[11] Recently, Vanne et al.[12] developed optimized MD schemes by analyzing the RD-complexity characteristics of HEVC interprediction.

The complexity control methods based on the ICC condition for both H.264 and HEVC can be classified as follows: Kim et al.[13] analyzed the encoding procedure and skipped several MBs to maintain the frame rate. A frame-level rate control scheme was proposed for HEVC based on support vector machine.[14] Zhang et al.[15] proposed a hierarchical complexity allocation scheme among frames and CUs. In Ref. 16, Zhao et al. determined the number of CU-splitting from each CU depth for complexity control.

This work focuses on software encoding, in particular, for low-delay P-frame (LDP) configuration of HEVC. The proposed algorithm can facilitate the video encoding by cooperating with the built-in hardware video coding functions on devices. The proposed complexity control mechanism is among the group of pictures (GOP) layer, frame layer, and the CU layer. Furthermore, the prediction unit (PU) encoding procedure is reordered. Experimental results showed that the proposed method could maintain the transmission quality effectively under both the ECC and ICC.

This paper is organized as follows: Sec. 2 describes the proposed complexity control method. The experimental results are described in Sec. 3. Finally, Sec. 4 presents the conclusion.

2 Complexity Control Method

This study explores the allocation of the computational complexity to each coding layer of HEVC for complexity control. Methods for allocating are detailed in the following subsections. First, the complexity for each GOP is equally distributed, and frames in the same position of the GOP are allocated to the same complexity. Second, the predicted mean absolute difference (MAD) value of each of the largest coding units (LCU) is adopted for the distribution of the complexity to each of the LCUs, and the maximal coding depth of each CU is estimated according to the information of the temporal or spatial correlation. Finally, the PU encoding procedure is reordered on the basis of the coding gain performance.[8]

2.1 *Group of Pictures and Frame Layers Complexity Allocation*

This study focuses on the LDP configuration in the HEVC encoder.[17] The LDP configuration consists of one I frame, with the rest being P frames. Figure 1 shows the LDP configuration, where GOP^n represents the n'th GOP. Each GOP is composed of four P-frames. For each GOP, the quantization parameter (QP) settings for the first and third frames are

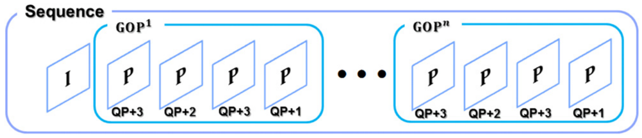


Fig. 1 An LDP configuration.

QP + 3, and those for the second and fourth frames are QP + 2 and QP + 1, respectively.

Accordingly, the complexity of all P frames, C_{T}^{P}, can be expressed as follows:

$$C_{\mathrm{T}}^{P} = C_{\mathrm{E}} - C^{I}, \tag{1}$$

where C_{E} is the total complexity and C^{I} is the complexity of the I frame. In this study, the complexity of the I frame was estimated according to the average complexity of the I frame from the five test sequences.

To estimate the computational complexity in the GOP layer, five test sequences from different classes were simulated. The details of the simulation environment are presented in Table 1. The reference software for HEVC was HM 12.1, and eight QP settings were selected. For each sequence, 97 frames, including the first I frame and 24 GOPs, were encoded. The computational complexity was determined according to the CPU clock. Sequences of Class D were excluded because of their low resolution.

A test sequence, Class C_BasketballDrill, with a QP of 32 was used as an example to show the computational complexity of frames. Figure 2 shows the complexity consumption of each frame, measured by CPU clock, which could be treated as a periodic signal with a period of four, except for the first GOP. In other words, processing each GOP involved almost the same complexity. Figure 2 also indicates that processing the fourth frame in each GOP involved more complexity compared with processing the other three frames. This is because the fourth frame was assigned the lowest QP, and it may increase the complexity.

Figure 2 shows the two important characters of the LDP configuration. The first one is that processing each GOP involved almost the same complexity for a test sequence under a specific QP setting. This character provides the

Table 1 Simulation environment.

Reference software	HM 15.0
Sequence	Class A_Traffic (2560 × 1600, 30 fps) Class B1_Kimono1 (1920 × 1080, 24 fps) Class B2_BQTerrace (1920 × 1080, 60 fps) Class C_BasketballDrill (832 × 480, 50 fps) Class E_Johnny (1280 × 720, 60 fps)
Frames to be encoded	97
Configuration file	Low delay P (IPPP...)
QP	20, 22, 25, 27, 30, 32, 35, 37
Hardware	PC with Intel i7 CPU (8 cores) 3.4 GHz, and 4 GB of RAM

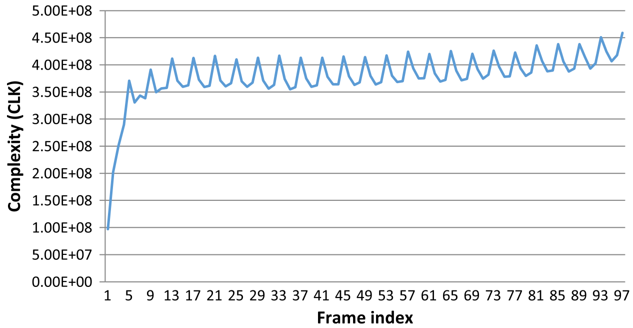


Fig. 2 Frame complexity consumption in Class C_BasketballDrill under a QP of 32.

estimation of each GOP complexity, C_{GOP}, which can be expressed as follows:

$$C_{\mathrm{GOP}} = C_{\mathrm{T}}^{P}/N_{\mathrm{GOP}}, \tag{2}$$

where N_{GOP} is the number of the GOP in the sequence.

The second character is that four frames composed a GOP according to the LDP configuration. Therefore, only the complexity of four frames had to be estimated. The frame complexity estimation can thus be simplified to only estimate the frame complexity from four different positions of the GOP, respectively. However, for each test sequence, the complexity consumption depends on QP values.

Let P_i denote the complexity ratio of the i'th position frame to the GOP, where the value of i is 1, 2, 3, or 4. Each P_i was calculated according to the complexity consumption of the individual frame from five test sequences with eight QP settings. For other QP settings, a linear interpolation was applied. Finally, the estimated ratios for frame layer allocation are plotted in Fig. 3. It shows that for low QPs, the effect of the complexity consumption for each frame on the QP factor is large. However, for high QPs, the rates for the first three frames are similar, and the rate for the fourth frame is high.

2.2 *Coding Unit Layer Complexity Allocation*

The CU layer complexity allocation is to distribute the complexity from each frame. In the proposed method, a frame contains only one slice and a slice is partitioned into many LCUs. According to the HEVC coding procedure, each LCU performs prediction, transformation, and quantization. After all LCUs in a slice are encoded, sample adaptive offset (SAO) and a deblocking filter (DBF) are applied to improve the frame quality.

The complexity of the slice, C_{SL}, can thus be expressed as follows:

$$C_{\mathrm{SL}} = C_{\mathrm{F}} - C_{\mathrm{SAO}} - C_{\mathrm{DBF}}, \tag{3}$$

where C_{F}, C_{SAO}, and C_{DBF} are the complexity of the frame, SAO, and DBF, respectively. Because the complexity use of

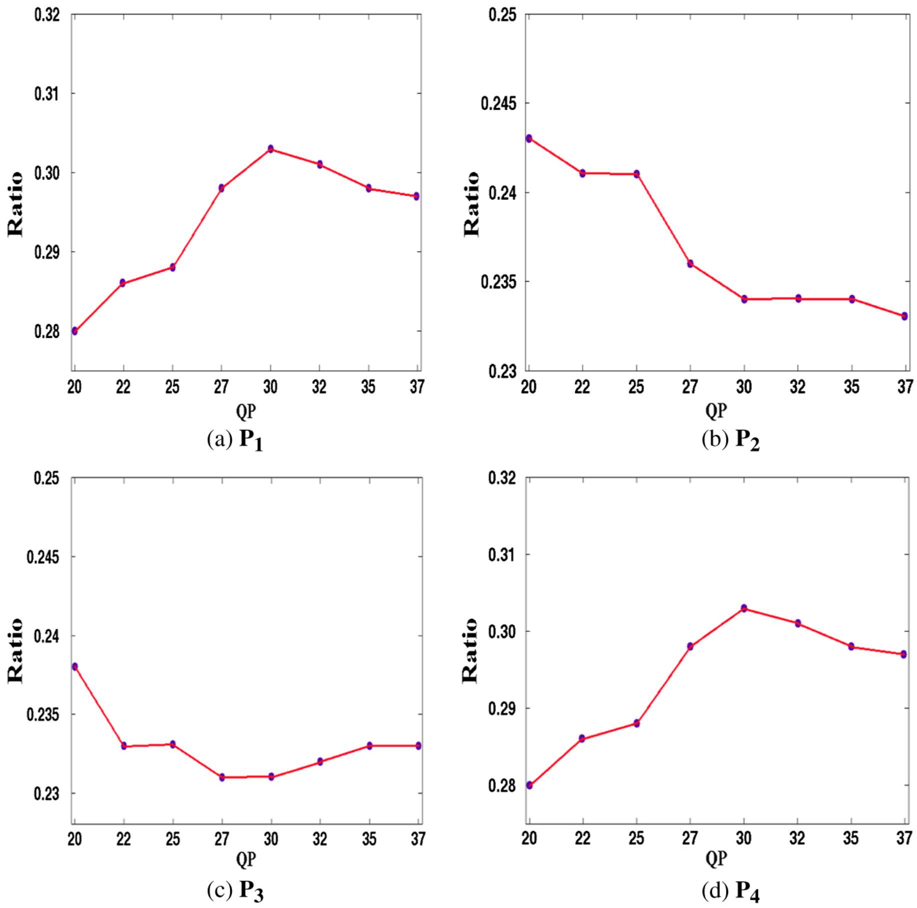


Fig. 3 Percentage of the average frame complexity within a GOP under different QP settings: (a) P_1, (b) P_2, (c) P_3, and (d) P_4.

the SAO and DBF coding tools did not show large fluctuations in response to different sequence contents and QP settings, for simplicity, the complexity of these two coding tools was considered constant and could be calculated in advance.

The complexity allocation to the LCU and the depth 1 of CU are described in the following subsections.

2.2.1 *Complexity allocation to largest coding units*

To efficiently allocate C_{SL} among the LCUs in a frame, a metric for estimating the computational complexity of an LCU is included. According to the coding procedure of HEVC, an LCU is composed of, at most, four CU coding depths, and its complexity is proportional to the number of coding depths. The optimal coding depth of each CU mainly depends on the video content and QP setting. Thus, in this work, the distribution of each LCU computational complexity is based on its content complexity estimation.

The MAD is the average difference between the current encoding pixel and its predicted pixel, estimated by the motion prediction within a block, and it is usually applied to estimate the content complexity of a block. In particular, the HEVC rate control algorithm employs a linear model to estimate the MAD of the current encoding frame based on the MAD of its previous frame[18] because of the strong correlation between successive frames. Therefore, in this work, the MAD of the colocated LCU in the previous frame was adopted to estimate the complexity of the current encoding LCU, and the complexity allocated to each LCU was proportional to the MAD of the colocated LCU in the previous frame.

The MAD for an LCU is calculated according to the following expression:

$$\mathrm{MAD} = \left(\sum_{i=0}^{2N-1}\sum_{j=0}^{2N-1} |a_{i,j} - \hat{a}_{i,j}|\right) / (2N \times 2N), \tag{4}$$

where $N = 32$, and $a_{i,j}$ and $\hat{a}_{i,j}$ are (i, j) pixels in the current LCU and predicted LCU, respectively. The allocated complexity for the i'th LCU in the current frame, $C_{\mathrm{LCU_i}}$, is expressed as follows:

$$C_{\mathrm{LCU_i}} = \left(\mathrm{MAD}_i / \sum_{i=1}^{N_{\mathrm{LCU}}} \mathrm{MAD}_i\right) \times C_{\mathrm{SL}}, \tag{5}$$

where N_{LCU} is the number of LCUs in the slice and MAD_i is the MAD value of the i'th LCU in the previous frame.

Class A_Traffic was used as an example for demonstrating LCU complexity allocation. Figure 4 shows the complexity consumption for the first 100 LCUs in the 12th slice with QP 32. The blue (solid) line shows the original complexity consumption, and the orange (dotted) line shows the complexity allocation based on the proposed equation in Eq. (5). The allocation performed using the proposed method was close to the original complexity consumption.

2.2.2 *Complexity allocation to the depth 1 of coding unit*

The proposed complexity allocation from the LCU to the depth 1 of a CU can be separated into two steps. In the first step, it is determined whether the search for the optimal

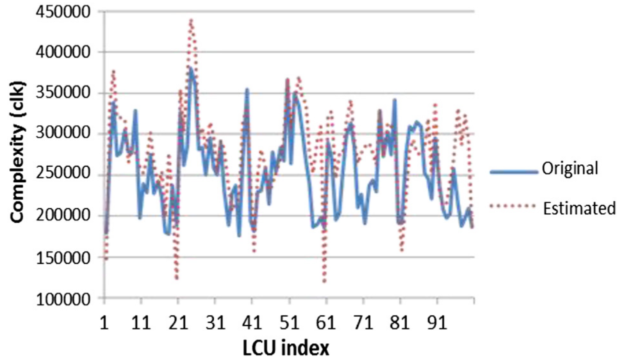


Fig. 4 Complexity consumption and estimation for the first 100 LCUs.

PU in depth 0 of the CU should be skipped. In the second step, the complexity is distributed among the four CUs in depth 1. Since each CU complexity is proportional to its maximal coding depth, the coding depth prediction for the current encoding CU is crucial.

The predicted LCU depth is first considered to determine whether the search for the optimal PU in depth 0 of the CU should be skipped. This is because if most of the CUs in the colocated LCUs are in large coding depths (e.g., depth 3), the search for the optimal CU depth is unlikely to yield depth 0. The computational complexity of the CU in depth 0 can thereby be reduced.

Coding depths from temporal and spatial colocated LCUs are employed. The coding depth of the colocated LCUs in the previous frame is denoted as $d_{\mathrm{LCU}}^{\mathrm{Co}}$ and the coding depths of the top LCU and left LCU are denoted as $d_{\mathrm{LCU}}^{\mathrm{Top}}$ and $d_{\mathrm{LCU}}^{\mathrm{Lef}}$, respectively. The complexity of the predicted coding depth of the current encoding LCU, $\hat{d}_{\mathrm{LCU}}$, is the average of these three coding depths as follows:

$$\hat{d}_{\mathrm{LCU}} = (d_{\mathrm{LCU}}^{\mathrm{Co}} + d_{\mathrm{LCU}}^{\mathrm{Top}} + d_{\mathrm{LCU}}^{\mathrm{Lef}})/3. \tag{6}$$

Simulation results indicated that if $\hat{d}_{\mathrm{LCU}}$ was >2, then the computational complexity of the CU in depth 0, denoted as $C_{\mathrm{CU}}^{D_0}$, could be saved.

The predicted CU depth is then used to determine the complexity allocation to the four CUs in depth 1. After the CU in depth 0 is encoded, the remaining complexity, $C_{\mathrm{LCU}} - C_{\mathrm{CU}}^{D_0}$, is distributed to the four CUs in depth 1. The complexity allocated to each CU is proportional to the predicted CU depth. Let $\hat{d}_j$ denote the predicted coding depth (predicted according to the colocated CU in the previous frame) for the j'th CU in depth 1, where j has the value 1, 2, 3, or 4. The complexity allocated to the j'th CU in depth 1, $C_{\mathrm{CU}}^{D_{1,j}}$, is expressed as follows:

$$C_{\mathrm{CU}}^{D_{1,j}} = w_j (C_{\mathrm{LCU}} - C_{\mathrm{CU}}^{D_0}), \tag{7}$$

where j represents the j'th CU in depth 1 and w_j is a weighting factor calculated as follows:

$$w_j = \hat{d}_j / \sum_{j=1}^{4} \hat{d}_j. \tag{8}$$

If four colocated CUs are in depth 0, then the depths of these CUs are zero. Under this condition, the remaining complexity, $C_{\mathrm{LCU}} - C_{\mathrm{CU}}^{D_0}$, is equally distributed among the CUs in depth 1. The complexity allocated to the CU in depth 1 can be continually allocated to its descendant CUs in depths 2 and 3 if the complexity is not completely consumed. In the proposed method, the complexity of the CU in depths 2 and 3 is not controlled.

2.3 *Modifying the Prediction Unit Coding Procedure*

HEVC employ coding tools for achieving high coding efficiency. These coding tools are applied in order under the encoding procedure. However, as the complexity is limited, coding tools with low coding gains can be omitted.

At a constant bit rate, the coding gain is defined as follows:[8,19]

$$\mathrm{CG} = \Delta\mathrm{PSNR}/\Delta\mathrm{Complexity}, \tag{9}$$

where ΔPSNR and ΔComplexity denote the amount of increase in the peak signal-to-noise ratio (PSNR) and complexity, respectively. Equation (9) indicates that if a coding tool is added to the encoding process, the ratio between the increased PSNR and the increased complexity is the coding gain for the coding tool. This is because the increase in both the PSNR and complexity is caused by the coding tool. Coding tools with higher coding gains should be processed earlier under the complexity constraint if the encoding procedure can be modified.

Coding gains for PU partitions were determined for the simulation environment presented in Table 1. The PU partitions included Inter $2N \times 2$, Inter $2N \times N$, Inter $N \times 2N$, Inter $N \times N$, AMP, Intra $2N \times 2N$, and Intra $N \times N$. Inter/Intra $2N \times 2N$ were set as the basic modes. Five test sequences were simulated, and for each test sequence, eight QPs were considered. For each test sequence, the coding gain for four PU partitions was determined. The average values of the coding gain for the five test sequences are listed in Table 2. The coding tool of Intra $N \times N$ had the highest coding gain and should receive the highest priority for being processed. By contrast, the coding tool of AMP had the lowest coding gain, and it could be skipped if the complexity was to be limited.

A flow chart of the modified PU coding procedure is shown in Fig. 5. The Intra $2N \times 2N$ procedure was moved to the front, immediately after the Inter $2N \times 2N$ procedure, and the AMP procedure was the last. To further reduce the complexity, either intraprediction or interprediction procedures can be included in the PU coding procedure, and this determination is based on the RD cost of Intra $2N \times 2N$ and Inter $2N \times 2N$. In the interprediction procedure, if

Table 2 The coding gain for PU partitions.

PU partition	Average CG (dB/kCLK)	Order
Intra $N \times N$	7.59×10^{-07}	1
$N \times 2N$	6.36×10^{-08}	2
$2N \times N$	4.28×10^{-08}	3
AMP	4.15×10^{-08}	4

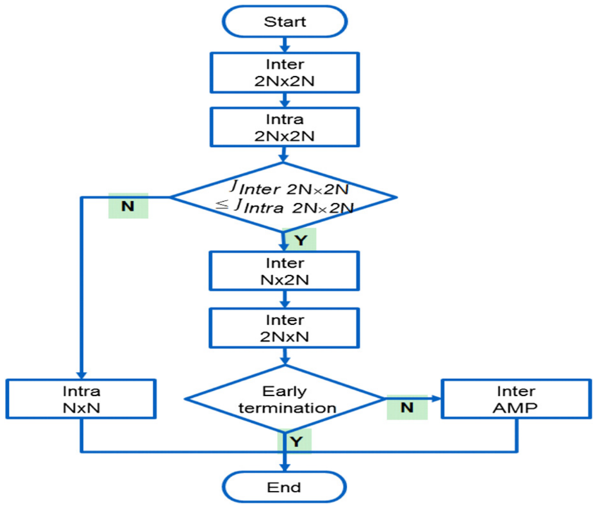


Fig. 5 Modified coding procedure for PU prediction.

the complexity is exhausted, then the AMP procedure can be skipped. Finally, Fig. 6 summarizes the proposed method.

3 Experimental Results

Simulations were designed to show the RD performance under the ECC and under both the ECC and ICC. The target parameter was set to 80% or 60%. The target parameter is defined as the percentage of the maximum possible complexity that can be used to encode a predefined segment of a video sequence.[11,15] In practice, for real-time complexity control, the total computational complexity can be estimated by the complexity consumption of an unconstrained GOP times the number of GOPs to be played. 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

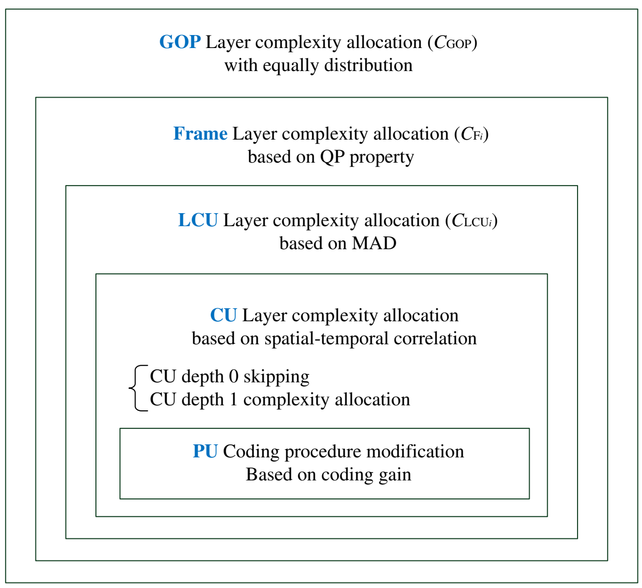


Fig. 6 Summary of the proposed coding-layered complexity allocation.

imposing any complexity constraint.[11] The target complexity, expressed as percentages of the total complexity, was then determined beforehand to evaluate the performance of the proposed method.

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

$$\mathrm{CCE}_{\mathrm{ECC}}(\%) = [(C_{\mathrm{pro}} - C_{\mathrm{T}})/C_{\mathrm{T}}] \times 100, \tag{10}$$

where C_{T} is the target complexity, and C_{pro} is the actual consumed complexity determined using the proposed method. The CCE for ICC was defined as follows:

$$\mathrm{CCE}_{\mathrm{ICC}}(\%) = \left(\sum_{i=1}^{N_f} \frac{|C_{\mathrm{F}_i} - C_{\mathrm{FM}} \times \mathrm{TP}| / C_{\mathrm{FM}} \times \mathrm{TP}}{N_{\mathrm{f}}} \right) \times 100, \tag{11}$$

where TP is the target parameter, N_{f} is the total number of frames, and C_{FM} is the maximal frame complexity selected from each frame of the original complexity.

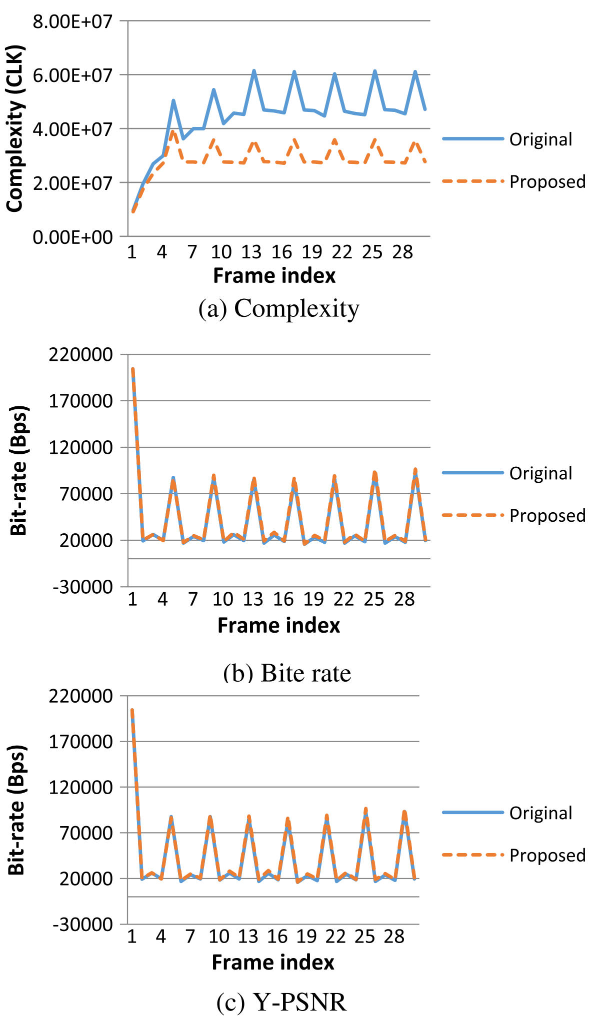


Fig. 7 Performances of Class C_RaceHorses from the first 30 frames under a QP of 32% and 60% of target complexity of ECC. (a) Complexity, (b) bit rate, and (c) Y-PSNR.

The simulation environment had the conditions shown in Table 1, except for the test sequence and QP setting. Thirteen test sequences were applied, and four QP settings were used: 27, 32, 37, and 42.

3.1 *Entire Complexity Constraint*

The first simulation was conducted to test the RD performance under the ECC. A test sequence, Class C_RaceHorses, with a QP of 32 was selected as an example to show the complexity consumption and the RD performance of each frame. Figures 7(a)–7(c) show the complexity, bit rate, and Y-PSNR performance for the first 30 frames, respectively. As shown in Fig. 7(a), the complexity consumption of each frame apparently decreased compared with the original complexity. Figures 7(b) and 7(c) show that the bit rate and the Y-PSNR performance approximated the performance of the original sequence for each frame, respectively.

Figure 8 shows the RD performance for the same test sequence. The RD performance decreased slightly as the percentage of the target complexity decreased. The RD performance is close to the RD performance without constraint. Table 3 lists the performances of the eight test sequences for the Bjøntegaard delta rate (BD-rate), Bjøntegaard delta peak signal-to-noise ratio (BD-PSNR),[20] and CCE for target complexities of 80% and 60% under the ECC. The sequence of B1_Kimono showed less deterioration in the BD-rate and BD-PSNR performance compared with other sequences. This is because B1_Kimono1 is a low-motion sequence, and the early termination in its prediction process, caused by the limited complexity, did not appreciably affect the performance. By contract, the drop in the BD-rate and BD-PSNR in high-motion sequences, such as A_Traffic and C_BasketballDrill, was larger than that in other sequences. As the target complexity dropped from 80% to 60%, the BD-rate increased and the BD-PSNR decreased. However, the BD-PSNR performance can maintain a decrease of <0.2 dB on average. Finally, the average CCE performance was only ∼0.3% or 0.35% for the target complexity of 80% or 60%, respectively, implying that the proposed method can effectively control the complexity under the ECC.

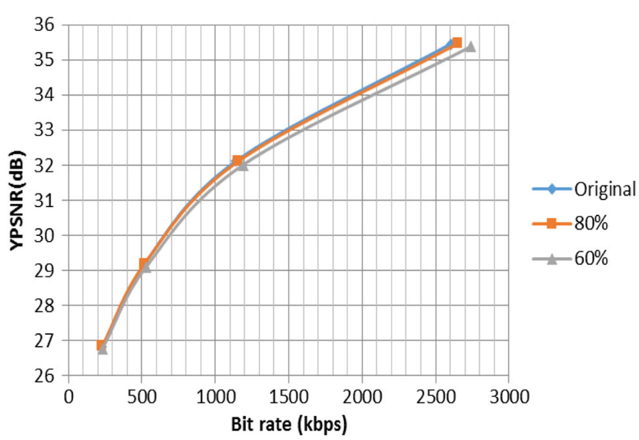
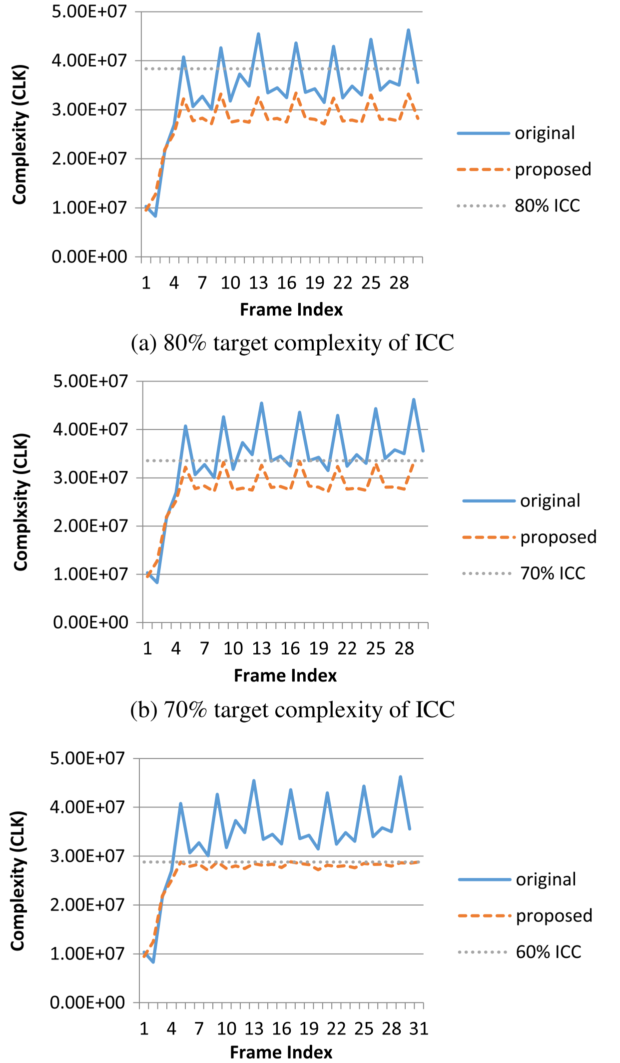


Fig. 8 RD performance of the Class C_RaceHorses under 80% and 60% of target complexities of ECC.

Table 3 Average BD-rate, BD-PSNR, and CCE performance under 80% or 60% of ECC.

Resolution	Video	Constraint	BD-rate (%)	BD-PSNR (dB)	CCE (%)
2560 × 1600	PeopleOn Street	80% ECC	−0.345	0.017	−0.4
30 fps		60% ECC	6.83	−0.316	0.54
2560 × 1600	Traffic	80% ECC	1.63	−0.058	0.54
30 fps		60% ECC	5.57	−0.192	0.58
1920 × 1080	Kimono1	80% ECC	−0.351	0.013	0.51
24 fps		60% ECC	1.02	−0.039	0.55
1920 × 1080	ParkScene	80% ECC	2.151	−0.064	0.52
24 fps		60% ECC	5.944	−0.171	0.58
1920 × 1080	BQTerrace	80% ECC	1.05	−0.024	0.51
60 fps		60% ECC	3.924	−0.079	0.52
1920 × 1080	Basketball Drill	80% ECC	0.978	−0.036	−0.25
50 fps		60% ECC	5.52	−0.28	−0.16
1920 × 1080	Cactus	80% ECC	0.671	−0.020	0.49
50 fps		60% ECC	3.588	−0.108	0.51
832 × 480	RaceHorses	80% ECC	1.74	−0.06	0.24
30 fps		60% ECC	8.92	−0.30	−0.20
832 × 480	Basketball Drill	80% ECC	0.978	−0.036	0.24
50 fps		60% ECC	6.821	−0.248	0.19
832 × 480	BQMall	80% ECC	3.859	−0.157	0.33
60 fps		60% ECC	13.88	−0.535	0.28
1280 × 720	FourPeople	80% ECC	0.858	−0.039	0.43
60 fps		60% ECC	3.466	−0.148	0.44
1280 × 720	Johnny	80% ECC	0.596	−0.023	0.40
60 fps		60% ECC	2.708	−0.093	0.38
1280 × 720	KristenAnd Sara	80% ECC	0.964	−0.038	0.4
60 fps		60% ECC	3.177	−0.122	0.38
	Average	80% ECC	1.136	−0.04	0.304
		60% ECC	5.489	−0.20	0.353



(a) 80% target complexity of ICC

(b) 70% target complexity of ICC

(c) 60% target complexity of ICC

Fig. 9 Comparisons of the complexity consumption under 80% target complexities of ECC, and 80%, 70%, or 60% of ICC, respectively.

3.2 *Entire and Instant Complexity Constraint*

The second simulation was performed to show the RD performance under both the ECC and ICC. A test sequence with a QP value of 27, Class C_BasketballDrill, was used as an example to show the complexity consumption under both the ECC and ICC. The target complexity for the ECC was set to 80%, and the target complexity for the ICC was set to 80%, 70%, or 60%. Figure 9 shows a comparison between the original and the actual complexity consumption for the three different complexity constraints of the ICC. The gray line shows the complexity bound for each frame.

Because of the ECC, each frame complexity allocation could not reach the 80% target complexity of the ICC, as shown in Fig. 9(a). Figure 9(b) shows that the 70% target complexity of the ICC constrained the complexity allocation for some frames, especially for the fourth frame of each GOP. As the target complexity of the ICC was reduced to 60%, the complexity allocation for most frames was bounded by the ICC constraint, as shown in Fig. 9(c). The inconsistency in the target complexity of the ECC and ICC results from the ECC being defined by the average of the total complexity and the ICC being defined by the maximum complexity of each frame. To show the complexity for both the ECC and ICC, the target complexity was set to 80% of the ECC and 60% of the ICC for each test sequence.

Table 4 lists the BD-rate, BD-PSNR, and CCE performance under the ECC and ICC. The BD-rate drop was ∼1.9%, and the BD-PSNR drop was ∼0.1 dB. The performance according to these measures decreased slightly compared with the performance for only the 80% target complexity of the ECC, as shown in Table 3. In addition, the CCE for the ECC and ICC could be maintained at ∼0.3% and 3.7%, respectively. Table 4 shows that the proposed complexity control method can effectively maintain the video quality for the target complexities of both the ECC and ICC.

Table 4 Average BD-rate, BD-PSNR, and CCE performance under 80% of ECC and 60% of ICC.

Resolution	Video	BD-rate (%)	BD-PSNR (dB)	CCE (%) ECC	CCE (%) ICC
2560 × 1600 30 fps	PeopleOn Street	1.989	−0.095	−9.54	2.04
2560 × 1600 30 fps	Traffic	2.959	−0.104	−3.55	3.92
1920 × 1080 24 fps	Kimono1	−0.106	0.004	−2.05	4.08
1920 × 1080 24 fps	ParkScene	2.999	−0.088	−1.88	4.37
1920 × 1080 60 fps	BQTerrace	1.96	−0.04	−2.71	5.76
1920 × 1080 50 fps	Basketball Drive	−0.733	0.023	−3.09	4.98
1920 × 1080 50 fps	Cactus	1.439	−0.044	−2.57	4.60
832 × 480 30 fps	RaceHorses	1.408	−0.046	−4.23	6.99
832 × 480 50 fps	Basketball Drill	2.216	−0.083	−2.22	4.30
832 × 480 60 fps	BQMall	5.104	−0.207	−1.07	5.89
1280 × 720 60 fps	FourPeople	1.854	−0.081	−7.8	3.27
1280 × 720 60 fps	Johnny	2.026	−0.072	−9.72	2.95
1280 × 720 60 fps	KristenAnd Sara	2.170	−0.085	−4.96	3.08
	Average	1.945	−0.0706	−4.2607	4.32

For comparison, we simulated the algorithm proposed by Corrêa et al.[11] Class B2_BQTerrace was adopted as an example to show their method, plotted in Fig. 10. For better expression, we only plotted the first 27 frames. The *x*-axis is the frame number, and the *y*-axis is the complexity consumption. The green (solid) line represents the constraint flag, which has only two values, 0 and 1. Flag 0 represents the unconstrained frames, and flag 1 represents the constrained frames. The blue (dashed) line shows the frame complexity consumption without constraint, and the red (dotted) line shows the frame complexity consumption under the algorithm in Ref. 11. It shows that as the flag equals to 0, the red (dotted) line can approach the points of the blue (dashed) line. This is because these frames are unconstrained frames. However, as the flag turns to 1, the actual complexity is lower than the complexity without constraint. Corrêa's method used the CU depth of unconstrained frame to restrict the CU depth for constrained frames.

Three test sequences, Basketball, BQTerrace, and Cactus were applied for performance comparison. Experiments were tested based on the first 97 frames under 60% of ECC. The result was listed in Table 5. It shows that Ref. 11 only can reduce about 18% of ECC, and ours can reach the constraint. The performance of Corrêa's method can be explained by Fig. 7 in Ref. 11. This figure presents the evolution of the number of constrained frames (i.e., Nc) during the encoding process for the four target complexities. It shows that when tighter target complexities are specified, Nc is increased until the predicted complexity either fits into the target complexity or reaches its upper limit (i.e., the full frame rate, 50 fps). Therefore, 97 frames are not enough to reach the requirement of 60% of ECC.

As a result, we can reach the following conclusions for performance comparison between these two methods:

1. When tighter target complexities are specified, it takes more frames to achieve the constraint of ECC in Corrêa's method.

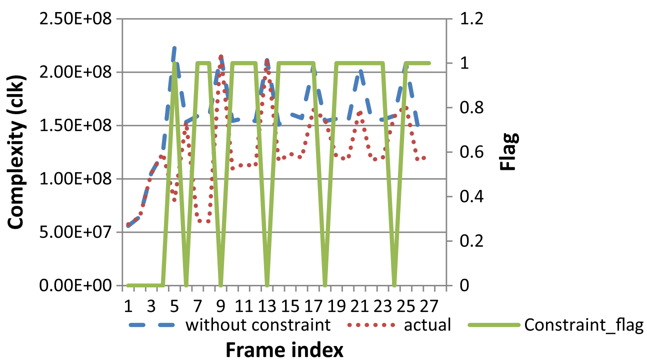


Fig. 10 An experimental result to show the algorithm of Ref. 11.

Table 5 Performance comparisons under 60% of ECC.

Constraint	Bitrate (%) Corrêa et al.[11]	Bitrate (%) Proposed	PSNR (dB) Corrêa et al.[11]	PSNR (dB) Proposed	Time saving (%) Corrêa et al.[11]	Time saving (%) Proposed
60% of C_E	−0.32	−0.51	−0.02	−0.10	17.82	39.95

2. Corrêa's method only can achieve complexity control under ECC, but ours can achieve complexity control under both ECC and ICC.

4 Conclusion

This study proposes a complexity control method for the LDP configuration in the HEVC encoder, focusing on software development. The proposed algorithm can facilitate the video encoding by co-operating with the built-in hardware video coding functions on devices. In the proposed method, the complexity was allocated among the GOP layer, the frame layer, and the CU layer. For the GOP layer, the complexity was equally distributed. For the frame layer, the complexity depended on the frame position in each GOP. The CU complexity allocation was proportional to its MAD value. Furthermore, the step in the PU encoding procedure was reordered according to the coding gain. By allocating the complexity to each coding layer of HEVC, the proposed method could simultaneously satisfy the ECC and ICC. Experimental results showed that as the target complexity under both the ECC and ICC was reduced to 80% and 60%, respectively, the decrease in the averaged PSNR was $\sim$0.1 dB with an increase of 1.9% in the BD-rate.

References

1. T. Wiegand et al., "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.* **13**(7), 560–576 (2003).
2. G. J. Sullivan et al., "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.* **22**(12), 1649–1668 (2012).
3. G. Corrêa et al., "Performance and computational complexity assessment of high-efficiency video encoders," *IEEE Trans. Circuits Syst. Video Technol.* **22**(12), 1899–1909 (2012).
4. Z. Li and G. Wen, "Reusable architecture and complexity-controllable algorithm for the integer/fractional motion estimation of H.264," *IEEE Trans. Consum. Electron.* **53**(2), 749–756 (2007).
5. C. S. Kannangara et al., "Complexity control of H.264/AVC based on mode-conditional cost probability distributions," *IEEE Trans. Multimedia* **11**(3), 433–442 (2009).
6. K. Lee, G. Jeon, and J. Jeong, "Fast reference frame selection algorithm for H.264/AVC," *IEEE Trans. Consum. Electron.* **55**(2), 773–779 (2009).
7. D. Martinez-Enriquez, A. Jimenez-Moreno, and 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), 826–834 (2010).
8. M. C. Chien, Z. Y. Chen, and P. C. Chang, "Coding-gain-based complexity control for H.264 video encoder," in *Proc. IEEE Int. Conf. Image Processing (ICIP '08)*, California, pp. 2136–2139 (2008).
9. J. Stottrup-Andersen, S. Forchhammer, and S. M. Aghito, "Rate-distortion-complexity optimization of fast motion estimation in H.264/MPEG-4 AVC," in *Proc. IEEE Int. Conf. Image Processing (ICIP '04)*, Singapore, Vol. 1, pp. 111–114 (2004).
10. Z. He et al., "Power-rate-distortion analysis for wireless video communication under energy constraints," *IEEE Trans. Circuits Syst. Video Technol.* **15**(5), 645–658 (2005).
11. G. Corrêa et al., "Complexity control of high efficiency video encoders for power-constrained devices," *IEEE Trans. Consum. Electron.* **57**(4), 1866–1874 (2011).
12. J. Vanne, M. Viitanen, and T. D. Hämäläinen, "Efficient mode decision schemes for HEVC inter prediction," *IEEE Trans. Circuits Syst. Video Technol.* **24**(9), 1579–1593 (2014).
13. W. Kim, J. You, and J. Jeong, "Complexity control strategy for real-time H.264/AVC encoder," *IEEE Trans. Consum. Electron.* **56**(2), 1137–1143 (2010).
14. J. Yu et al., "Frame level rate control based on support vector machine in HEVC," in *14th Int. Symp. Communications and Information Technologies (ISCIT '14)*, pp. 517–521 (2014).
15. Y. Zhang et al., "An optimally complexity scalable multi-mode decision algorithm for HEVC," in *Proc. IEEE Int. Conf. Image Processing (ICIP '13)*, Melbourne, VIC, pp. 2000–2004 (2013).
16. T. Zhao, Z. Wang, and S. Kwong, "Flexible mode selection and complexity allocation in HEVC efficiency video coding," *IEEE J. Sel. Top. Signal Process.* **7**(6), 1135–1144 (2013).
17. JCT-VC, "High efficiency video coding (HEVC) test model 15 (HM15) encoder description," in *17th JCT-VC Meeting JCTVC-Q1002*, Valencia (2014).
18. H. Choi et al., "Rate control based on unified RQ model for HEVC," in *ITU-T SG16 Contribution, JCTVC-H0213*, San José (2012).
19. C. Kim and J. Xin, "Hierarchical complexity control of motion estimation for H.264/AVC," Mitsubishi Electric Research Laboratories, TR2006-004 (2006), http://www.merl.com
20. G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves," in *ITU-T SG16 Q.6 Document, VCEG-M33*, Austin (2001).

Jiunn-Tsair Fang received the PhD degree in electrical engineering from National Chung-Cheng University, Taiwan in 2004. Currently, he is an assistant professor in the Department of Electronic Engineering at Ming Chuan University, Taiwan. His researching interest includes video/image coding, audio processing, and channel coding.

Kai-Wen Liang received the MS degree in communication engineering from National Central University (NCU). She is currently pursuing the PhD degree at the VAPLab in the Department of Communication Engineering at NCU, Taiwan. Her research interest is video coding.

Zong-Yi Chen received the BS degree in electrical engineering and the MS degree in communication engineering from National Central University (NCU), Taiwan, in 2005 and 2007, respectively. He is currently pursuing the PhD degree at the Video-Audio Processing Laboratory (VAPLab) in the Department of Communication Engineering at NCU, Taiwan. His research interests include video/image processing and video compression.

Wei Hsieh received the BS degree in electrical engineering and the MS degree in communication engineering from Yuan Ze University and NCU, Taiwan, in 2012 and 2014, respectively. His research interest is video coding.

Pao-Chi Chang received the BS and MS degrees from NCTU, Taiwan, and the PhD degree from Stanford University, California, 1986, all in electrical engineering. From 1986 to 1993, he was a research staff at IBM T.J.Watson Research Center, New York. In 1993, he joined the faculty of NCU, Taiwan, where he is presently a professor in the Department of Communication Engineering. His main research interests include speech/audio coding, video/image compression, and multimedia retrieval.